Institute Of Business Management

College of Computer Science & Information Systems (CCSIS)

FINAL YEAR PROJECT REPORT

Session 2021-2022

"Skin Cancer Detection Using Machine Learning & Deep Learning"

SUBMITTED BY

Hamza Majid (20181-24736)
Sarib Bin Nasir (20181-24042)
Ahsan Ali Khan (20181-24496)

Bachelors of Science
In
Computer Science

PROJECT SUPERVISOR

Eng. Muhammad Waqar Khan

ACKNOWLEDGMENTS

First and foremost, we would like to express our gratitude and thankfulness to Allah, the Almighty for bestowing upon us the greatest possible aptitudes to complete this project. Through His will, blessings and endowments we were able to complete our project successfully. We have taken efforts in this project. We are highly indebted to Eng. Muhammad Waqar Khan Professor of College of Computer Sciences and Information Technology, Institute of Business Management, for his guidance and constant supervision which tremendously aided us, as well as for providing necessary information regarding the project and also for his support in gathering information for this project. We are grateful to Eng. Muhammad Waqar Khan for helping us efficiently organize our project, including authoring this report his invaluable guidance, oversight, and direction on this project.

In addition, we'd like to express our gratitude to Dr. Tariq Rahim Soomro, the Dean of CCSIS and his crew for always supporting students in achieving their objectives. Also, it would not have been possible without the kind support and help of our teacher and organization. We would like to extend our sincere thanks to all of them.

Thank you.

You’re sincerely,
Hamza Majid (20181-24736)
Sarib Bin Nasir (20181-24042)
Ahsan Ali Khan (20181-24496)

ABSTRACT

As the time has passed we have observed that the cases of cancer are increasing day by day. Not a day in a year goes by when a death has not occurred due to the complication of cancer. Skin cancer is one of the most common cancer as it not limited to one gender or particular age, The problem with the death due to increase in complication is related to the late detection of the cancer or the overall cost for the treatment is expensive. Skin cancers such as basal cell carcinoma, squamous cell carcinoma, and melanoma are becoming more prevalent.

The detection of this cancer is done by many methods but the thing which is common in them is that they are expensive. And many people who might don't have skin cancer there skin problem can be due to sensitivity or rash or allergy, due to them they delay the process hence making it critical for the patients who needed attention on time.

We can incorporate a way to predict whether the patient is cancerous or not using the machine learning algorithm and Deep Learning Algorithms. Machine learning algorithm has come far beyond from their boundaries and limitations.

Because of their potential pattern recognition capabilities, deep learning (DL) models have gotten a lot of interest in medical imaging. However, Deep Neural Networks (DNNs) need a large quantity of data, and because there isn't enough data in this subject, transfer learning might be a good alternative. DNNs used for illness diagnosis are methodically focused on increasing prediction accuracy without giving a figure for prediction confidence. Gaining physicians' confidence and faith in DL-based solutions requires knowing how confident a DNN model is in a computer-aided diagnostic model. This paper addresses this problem by presenting three distinct strategies for assessing skin cancer detection uncertainty using photos. It also uses innovative uncertainty-related measures to assess and compare the performance of various DNNs. The findings show that predictive uncertainty estimation approaches can identify dangerous and incorrect predictions with a high uncertainty estimate.

We can predict whether someone has skin cancer or not by looking at thousands of images from other people and let the machine compare and predict.

This can help people in early detection and saving them time and money in future also there would be less congestion in the detection systems hence the numbers of critical patient can arrive much early hence saving there life.

Keyword: Deep learning, Machine Learning, Algorithm, Skin Cancer, Deep Neural Networks.

Contents

1. Introduction

1.1 Purpose of the Project

The cancer is the fastest disease in the world, which is responsible for taking many live every year. With the detection at the right time we can make the consequence less and prevent the spread of the disease. There are many medical method to identify whether the patient is suffering from cancer or not but the problem with them is that they all are expensive, so we need to identify a method to identify and predict cancer using the picture. So that it can be early detect with saving the expense and so that it can be expended on curing cancer. The purpose of the project was to detect the skin cancer by using pictures. We are designing a model which will detect the condition of the cancerous lump from picture that will help the people to diagnose the disease early with respect to the lumps size and scar of the disease .Each algorithm model was designed specifically for the purpose of the prediction of the disease by looking at the pictures. For testing the model right now we are using HAM1000 data-set and the images from that data set. IT contains ten thousand and 15 images with 7 types of skin lesion. [1]

1.2 Scope of the Project

Business Goals:

1. We can save the money by identifying the cancer using ML Deep learning algorithms.
2. This can help us save life by early detection which will prevent the disease form spreading in critical organs. Saving life by early detection
3. Money can be saved of both the healthcare facilities hence money and man power can be reduced or minimized .saving money and manpower
4. The time is saved hence the normal people spend many precious hours in the waiting lines of testing facilities hence this time can be saved.[2]

Technical Goals:

1. The proposed model have end-to-end structure without manual feature extraction and selection method. An end-to-end structure without manual extraction is how the model is supposed to work.
2. 10015 images of 7 types of skin lesion for the prediction of the skin cancer.
3. We show the result of the different algorithms which will be using for prediction of the cancer system and show the best accuracy rate among all of them.

   The pre-trained model has been proved to produce excellent results in three distinct dataset types.

1.3 Research Work

On the basis of our project, we did major research work. We collected and studied numerous articles, research papers, blogs and projects. Which we included in our literature review. In the process we analysis that every individual worked on different methodology including single algorithm, deep learning and convolutional neural networking concepts. As for our manual research we studied more than twenty research work and came to the conclusion to work on different machine learning and deep learning algorithms and provide the best prediction and classification solution. [3]

2. Requirement Analysis

2.1 Domain requirements

Domain of the project is that it can help in medicinal industry and department as the project will have the capability to scan the image and then compare it with other images of skin cancer patients using different algorithms.

2.2 Functional requirements

There are many kind of lumps from swelling caused by stress of a muscle, over extending of joint to cancerous lumps, so this would help user in identifying whether the lump is dangerous or not. The user would be able to identify cancerous ailments by lending his pictures into the system consisting of ML and Deep learning algorithms. This would help the individual in identifying whether he should worry on the lumps and what kind of any further treatment should be done depending upon the prediction.

2.3 Non-functional requirements

The project would be portable, initially the back-end system would be created which will be portably ready that it can easily be integrated with GUI so that it can easily be operated by people with minimum computer knowledge.

It would ensure that the images in its possession is safe and confidential. The project would predict with reliability because someone's life may depend on it hence we would make sure that this system predicts according to the true signs of tumor comparing with other images.

At least 5 algorithms would be used to make sure that the prediction is accurate to the condition of tumor lump. We would try to make the system faster so that it can predict the illness at faster rate. [5][6][7]

3. Literature review

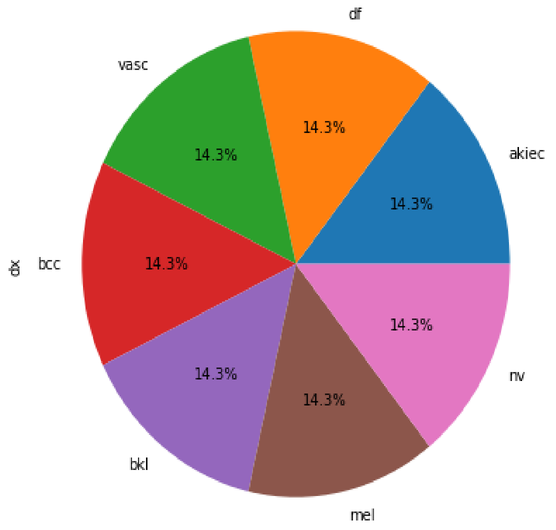
S.No	Title	Methods	Finding/Result
1.	DETECTION OF SKIN CANCER USING CNN ALGORITHM	Dermoscopic images were used to train and test the Inception v4 CNN architecture.	Used HAM1000 data for detection of skin cancer. Furthermore they use transfer learning to improve the accuracy. For transfer the learning MED-NODE dataset is used, using ResNet Model. Using transfer learning the accuracy has been improved to 90.51%.
2.	Convolutional Neural Networks for classifying skin lesions	CNN model has 4 layers with 2 conv layer in first 2 layers and 3 conv layer in last 2 layers with 64, 128, 256, 521, 512 neuron in each layer respectively. Also have 2x2 max pooling. In last they have 3 Fully connect layer with 4096 neurons reach.	The model for 50 epochs it was observed that a test accuracy of 78 percent was observed. It was observed that the accuracy for both training and test data kept increasing up to 30 epochs and gradually started saturating for the later epochs. Similarly, the training and test loss decreased exponentially for the first 30 epochs and remained unchanged for the next epochs.
3.	Melanoma skin cancer detection using deep learning and classical machine learning techniques: A hybrid approach	CNN model is used along with 2 machine learning algorithm KNN and SVM, and majority voting. CNN model have 3 hidden layer with 2x2 max pooling layer and dropout layer of 0.4	The CNN model has been trained on 10 epochs with the accuracy of 85.5%, other 2 machine learning algorithm have a bit lower accuracy as KNN accuracy has been observed with around 57.3% and SVM accuracy has been observed with 71.8%. The most accurate

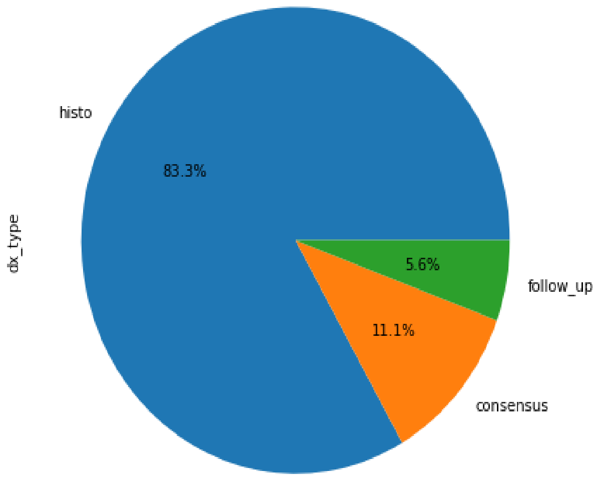
		before and after flatten layer. In KNN it use 5 nearest neighbor.	prediction has been made by majority voting algorithm with accuracy rate of 88.4% which is the highest with compare to others.
4.	Disease Classification based on Dermoscopic Skin Images Using Convolutional Neural Network in Teledermatology System	Also in this research CNN model is used but the dimension of the model and how many layers are used are not mention.	Using Convolutional Neural Networks, we offer a skin illness classification system based on dermoscopic skin images (CNN). Inception and Mobilenet v1 are two pre-trained CNN models that we tested. Melanocytic Nevi (nv) has the most dermoscopic image data, with 5,954 images, and so becomes the skin disease class with the highest proportion of right predictions on the Inception V3 model, at 90%. Dermatofibroma (df), which contains the fewest dermoscopic image data (109 images), is the skin disease class with the lowest percentage prediction (67 percent in the MobileNet v1 model and 17 percent in the Inception V3 model). With an original data amount of only 131 images, vascular has an accuracy rate of 70% on the MobileNet v1 model and 90% on the Inception V3 model. Based on the results of testing the application of the CNN model on web classification, the

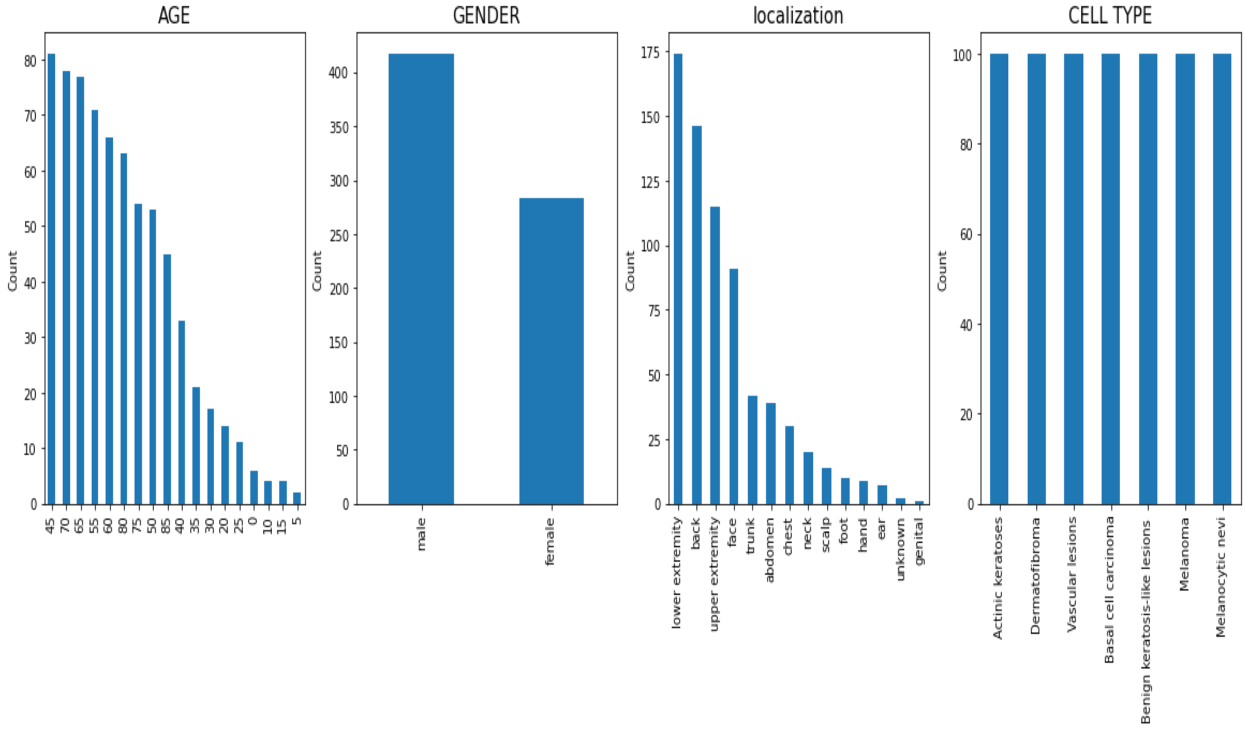
			disease classification system can be applied in Teledermatology applications.
5.	Confidence Aware Neural Networks for Skin Cancer Detection	The CNN model is also employed in this study, although the size of the model and the number of layers used are not specified.	The CNN model has been trained on 10 epochs with the accuracy of 83.5%.
6.	Deep learning design for benign and malignant classification of skin lesions: a new approach	The model use in this research is Support Vector Machine (SV M) and it uses ResNet50 per train model to better accuracy.	It is observed that the proposed technique of employing ResNet50 hybridized with SVM achieves the best performance, specifically with the ISIC2017 dataset, producing 99.19% accuracy, 99.32% area under the curve (AUC), 98.98% sensitivity, 98.78% precision, 98.88% F1 score and 2.6988 s computational time.

4. Data Set Details

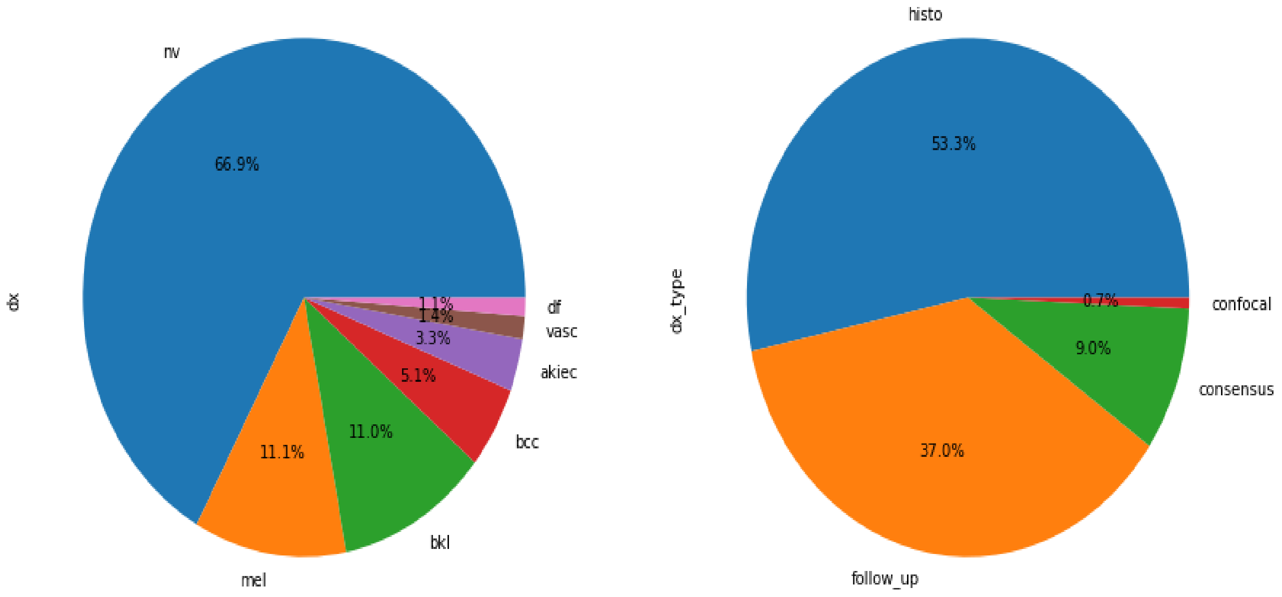
4.1 Augmented Data (Decreased)

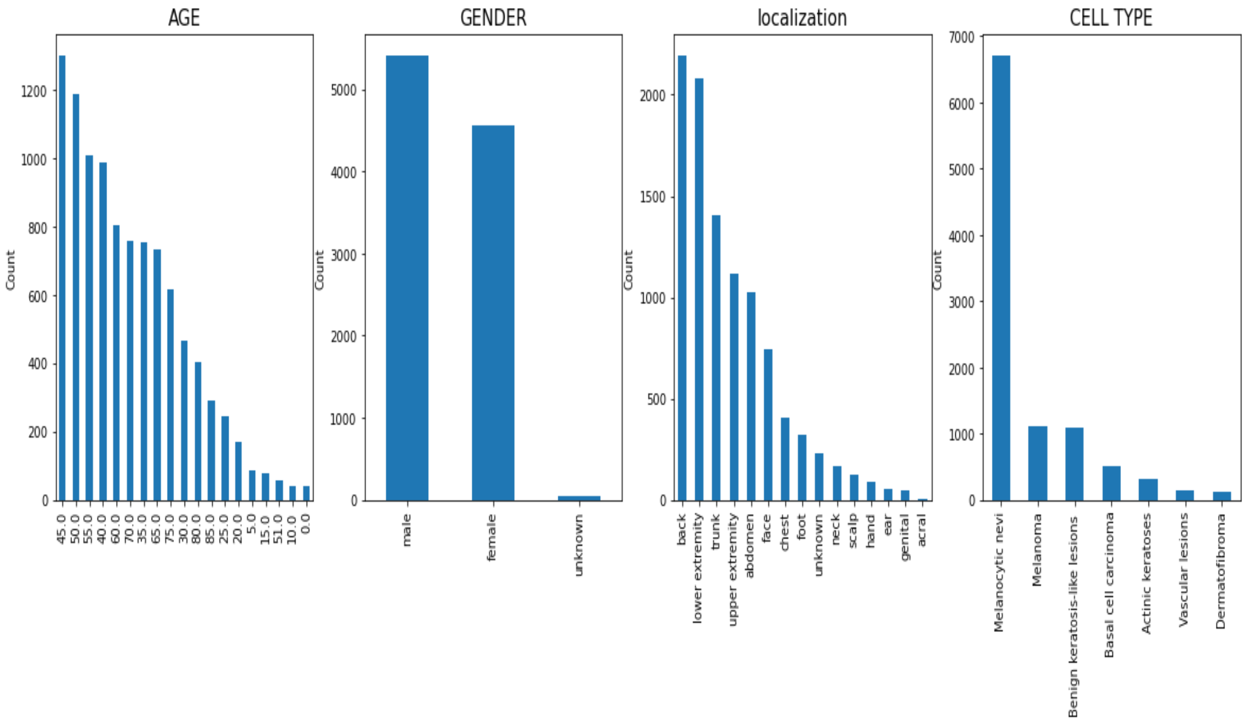




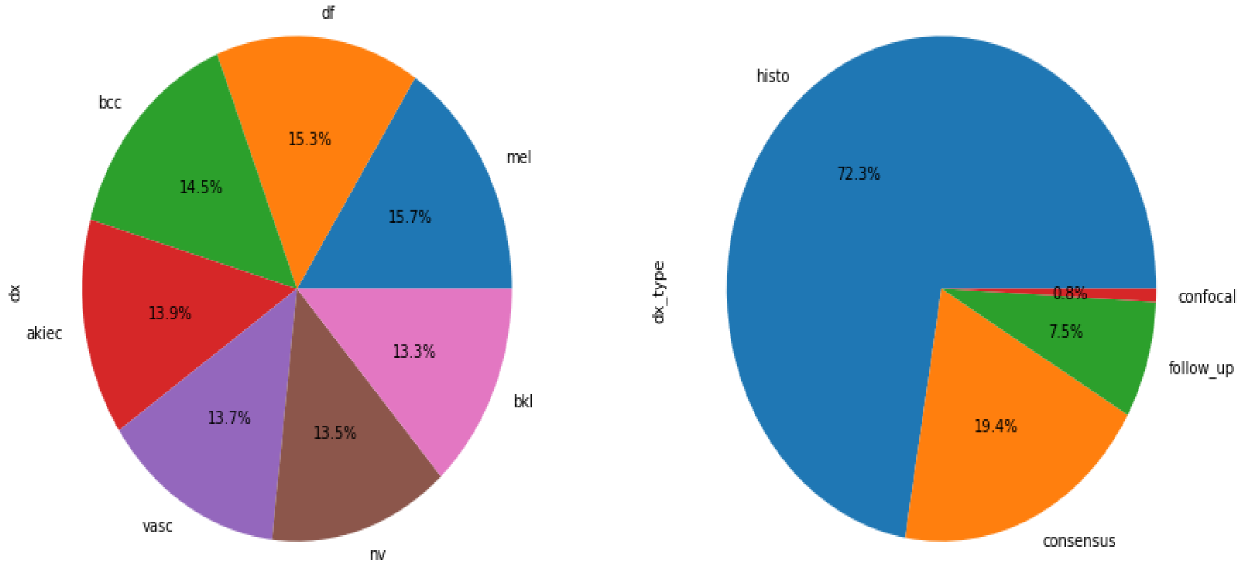


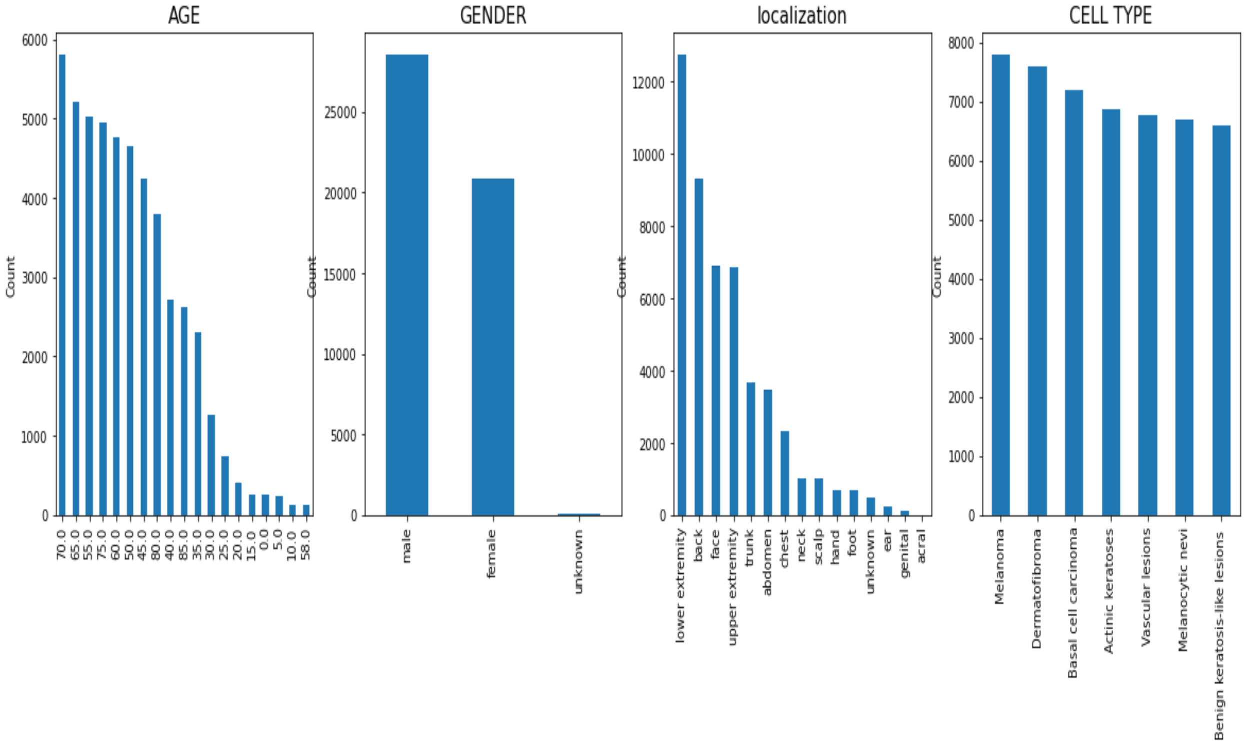
4.2 Original Data





4.3 Augmented Data (Increased)





5. Tools and Techniques

This project is based on machine learning and deep learning algorithms that will predict whether a person has a cancer or not. To apply these techniques we will be using Python as Programming language. Python is a high-level, general-purpose programming language that is interpreted. The use of considerable indentation in its design philosophy promotes code readability. Its language elements and object-oriented approach are aimed at assisting programmers in writing clear, logical code for both small and large-scale projects.[9][10][11]

Python can be used for various purposes

1. Mathematical computation using Numpy
2. Pandas library can manipulate data
3. TensorFlow, PyTorch, SciKit Learn are used for training machine learning models and deep learning models
4. Data visualization using Matplotlib and seaborn
5. Django framework for web development

Python is ideal for machine learning applications because of its advanced features and libraries such as tensorFlow, Keras, and Sciket Learn, which can be used to implement different algorithms and use those methods for training and testing datasets. The powerful Python environment is also capable of processing large datasets and performing complex calculations. [10][11]

1. NumPy

Large, multi-dimensional arrays and matrices are supported by NumPy, a library for the Python programming language, along with a substantial number of high-level mathematical operations that may be performed on these arrays. Jim Hugunin originally developed Numeric, the predecessor to NumPy, with assistance from a number of other programmers. Travis Oliphant developed NumPy in 2005 by heavily altering Numeric to incorporate capabilities of the rival Numarray. Numerous people have contributed to the open-source programme NumPy. A project sponsored financially by NumFOCUS is NumPy.

2. Pandas

Pandas is a data analysis and manipulation software package created for the Python programming language. It includes specific data structures and procedures for working with time series and mathematical tables. It is free software distributed under the BSD license's three clauses. The word is derived from "panel data," a phrase used in econometrics to refer to data sets that contain observations for the same persons throughout a range of time periods. Python data analysis is a play on words in the name of the thing. When Wes McKinney worked as a researcher at AQR Capital from 2007 to 2010, he began creating the pandas that would eventually become famous.

3. TenserFlow

A free and open-source software library for artificial intelligence and machine learning is called TensorFlow. Although it can be applied to many different tasks, deep neural network training and inference are given special attention.

The Google Brain team created TensorFlow for use in internal Google research and production. 2015 saw the maiden release under the Apache License 2.0. TensorFlow 2.0, the upgraded version of TensorFlow from Google, was launched in September 2019.

A large number of programming languages, including Python, Javascript, C++, and Java, can use TensorFlow. This adaptability allows for a wide range of applications across numerous industries.

4. Scikit-learn

A free machine learning library for the Python programming language is called Scikit-learn. Support-vector machines, random forests, gradient boosting, k-means, and DBSCAN are just a few of the classification, regression, and clustering algorithms it offers. It is also built to work with Python's NumPy and SciPy scientific and numerical libraries. A NumFOCUS fiscally sponsored project is scikit-learn.

5. Matplotlib

For the Python programming language and its NumPy numerical mathematics extension, Matplotlib is a graphing library. For integrating charts into programmes utilising all-purpose GUI toolkits like Tkinter, wxPython, Qt, or GTK, it offers an object-oriented API. It is not recommended to use the procedural "pylab" interface, which is based on a state machine (similar to OpenGL) and was created to closely mimic the MATLAB interface. Matplotlib is used by SciPy.

6. Keras

A Python interface for artificial neural networks is provided by the open-source software package known as Keras. The TensorFlow library interface is provided by Keras.

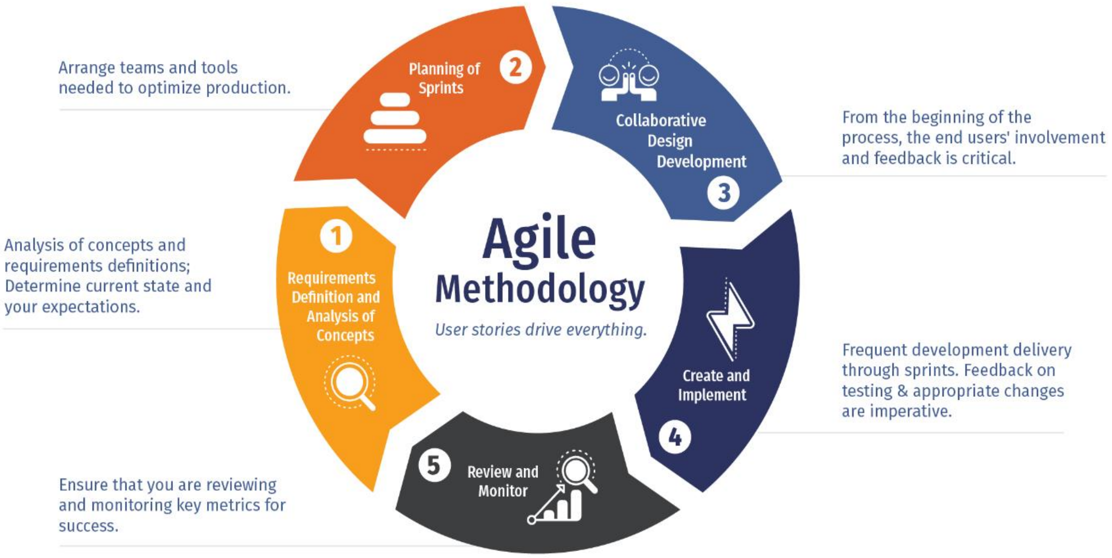
TensorFlow, Microsoft Cognitive Toolkit, Theano, and PlaidML were just a few of the backends that Keras supported up until version 2.3. One and only TensorFlow is supported as of version 2.4. Its user-friendliness, modularity, and extensibility are its main design goals as it aims to facilitate quick experimentation with deep neural networks. It was created as a component of the research project ONEIROS (Open-ended Neuro-Electronic Intelligent Robot Operating System), and François Chollet, a Google engineer, is its principal author and maintainer. The deep neural network model for Xception was also created by Chollet.

K
matplotlib
pandas
NumPy
scikit
learn

5. Tools Designing

5.1 Development Model

For this model we will use agile development model as requirements and solutions evolve through collaboration between self-organizing cross-functional teams. We can efficiently repeat all of our actions since this model follows the pattern of a recurrent loop and then release a development build. The integrity of this operational core is preserved while the system progresses through revisions. [12][13]


Agile
Methodology
User stories drive everything.
1
Requirements Definition and Analysis of Concepts
2
Planning of Sprints
3
Collaborative Design Development
4
Create and Implement
5
Review and Monitor

Arrange teams and tools needed to optimize production.

From the beginning of the process, the end users' involvement and feedback is critical.

Analysis of concepts and requirements definitions; Determine current state and your expectations.

Frequent development delivery through sprints. Feedback on testing & appropriate changes are imperative.

Ensure that you are reviewing and monitoring key metrics for success.

Fig: 5.1

5.2 Development life cycle:

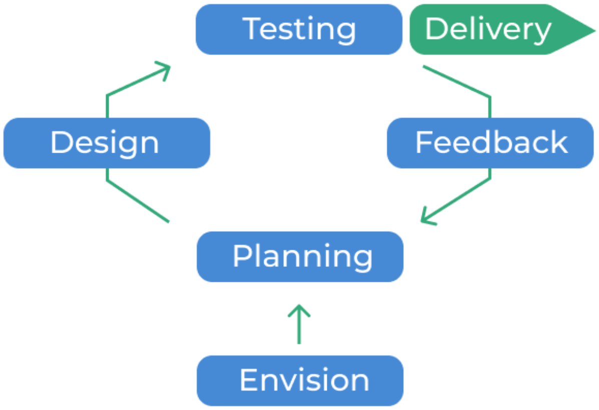
Agile methods or agile processes promote a disciplined project management process that encourages frequent inspection and adaptation, a leadership philosophy that encourages teamwork, self-organization, and accountability, a set of engineering best practices that allow for rapid delivery of high-quality software, and a business approach that aligns development with customer needs and company goals.

Any development approach that adheres to the Agile Manifesto's principles is referred to as agile development. The Manifesto was written by a group of fourteen software industry leaders, and it represents their knowledge of what techniques work and don't work in software development. [11][15]

5.3 Advantages and Disadvantages

Pros	Cons
When working with agile method we get more flexibility	Hard to predict
Product get to market faster as there are initial development releases	Final product is not released first because the development releases check whether there are any bugs or not
Better communication because with initial development release we can improve the function	Documentation gets left behind as there are continuous changes in the project

5.4 Agile Process Flow Chart



6. Phases of the Project

Because this project is on machine learning and related approaches, it employs a variety of machine learning algorithms to analyses the data set and produce correct results. This project entails the use of machine learning algorithms to handle the dataset, with two primary phases:

- Training phase
- Testing phase

6.1 Training Phase

The **training phase** entails obtaining a certain proportion of the dataset and running it through a training phase with the appropriate algorithm to train the algorithm for the kind of input to evaluate and give findings. The dataset input in the form of images will be used in the training phase of this project, and the training will be undertaken for assessing variables such as color, texture, and size in order to identify skin cancer.[16][17]

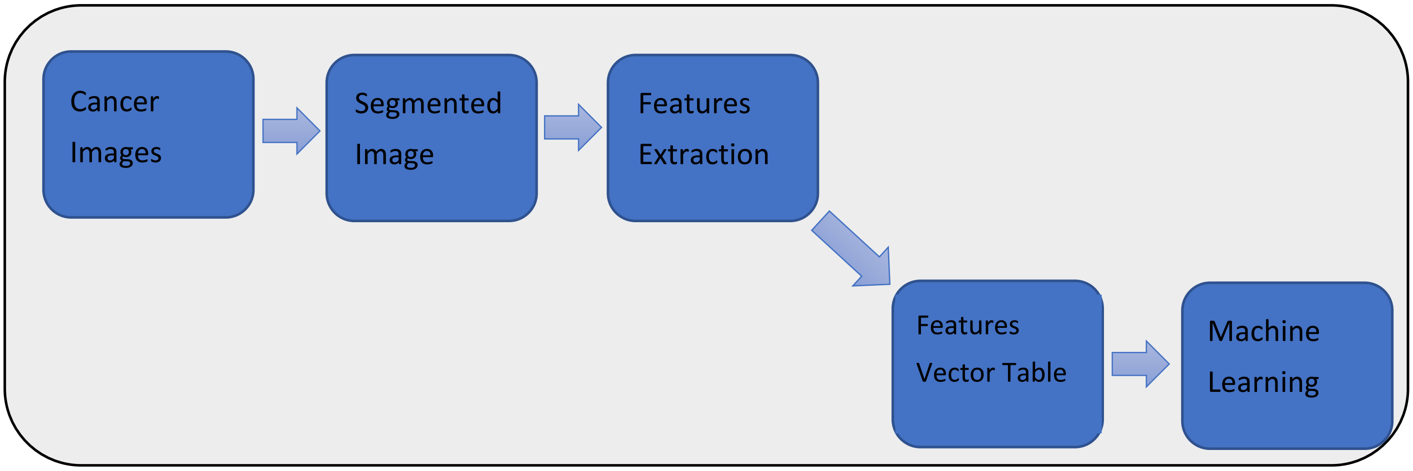


Fig: 6.1

6.2 Testing Phase

Similarly, the **testing phase** entails obtaining a specific proportion of the dataset and sending it through the testing phase of the algorithms to ensure that the generated output is accurate. Examining and evaluating characteristics inside the input from the dataset photos will also be -**part of the testing. These findings are then analyzed and compared in order to determine which method, out of the several that were employed in the project, is the most effective.[17][18][19]

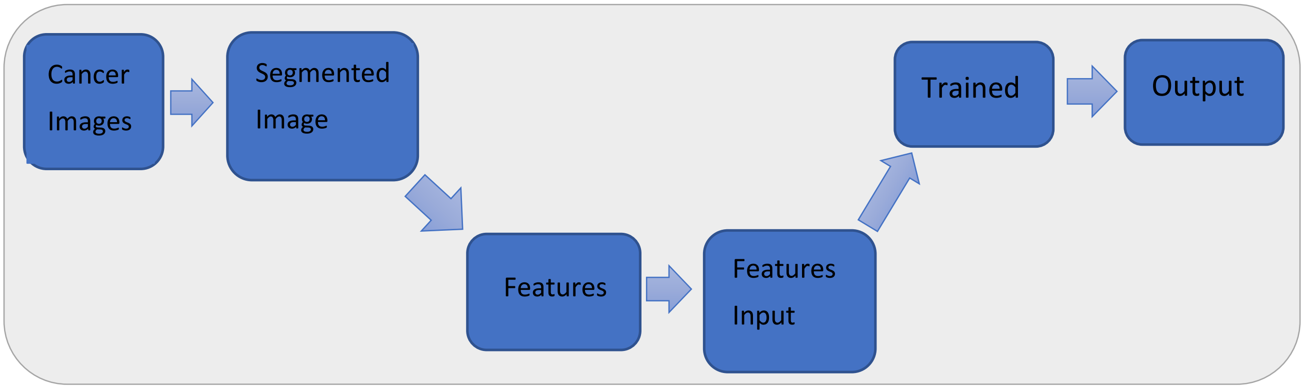


Fig: 6.2

6.3 Processing Steps in Training and Testing Phases

6.3.1 Step 1: Division of datasets[32][33]

Training set (75% of the selected data set): This is utilized to develop our prediction method and fine-tune the neural network's weights. Our algorithm attempts to adapt to the peculiarities of the training data sets. In this step, we'll put the four algorithms we chose to work in order to compare their results during Cross-Validation and for additional processing. Different parameter options are available for each type of algorithm (the number of layers includes in a Neural Network, the numerous trees in an arbitrary Forest, etc.). We attempted to choose the optimal solution for each algorithm.

Test set (25% of the selected data set): we will work on our favorite prediction algorithms in this step. As a result, we apply our selected prediction algorithm to our test set to see how it performs so that we can get a sense of how well it performs on unknown data.

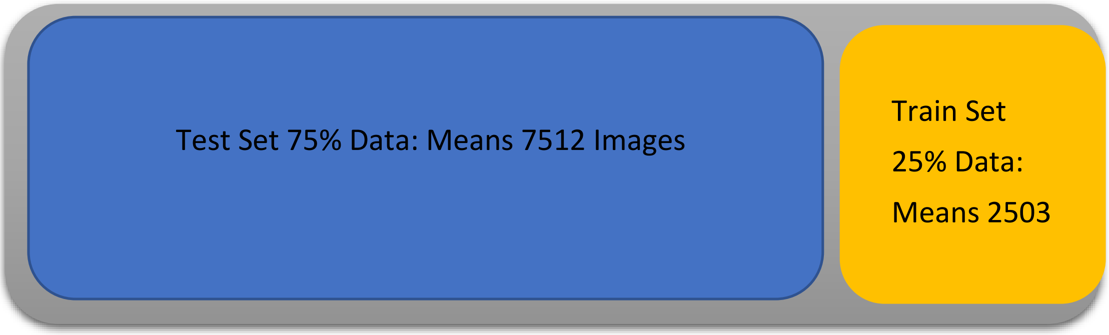


Fig: 6.3

6.3.2 Step 2: Feature Vector Table

The previous step's characteristics will be presented in the form of a huge table combining shape, texture, and color information in one file.

The two primary phases in picture preprocessing are image conversion and resizing. The initial stage is to read, analyses, and validate the image's dimensionality. A 3D picture must be converted to a 2D image if the image is 3D. Second, we scale the 2D image's dimensionality to (255 255). This stage's output will be utilized as the input for extracting the image's features.

7. Machine Learning & Deep Learning

7.1 Machine Learning

Machine learning is the study of computer algorithms that can learn and develop on their own with experience and data. It is considered to be a component of artificial intelligence. Machine learning algorithms create a model based on training data to make predictions or judgments without having to be explicitly programmed to do so. Machine learning algorithms are utilized in a wide range of applications, including medicine, email filtering, speech recognition, and computer vision, where developing traditional algorithms to do the required tasks is difficult or impossible.

However, not all machine learning is statistical learning. A subset of machine learning is strongly related to computational statistics, which focuses on making predictions using computers. The discipline of machine learning benefits from the study of mathematical optimization since it provides tools, theory, and application domains. Data mining is a similar branch of research that focuses on unsupervised learning for exploratory data analysis. Data and neural networks are used in some machine learning implementations to replicate the functioning of a biological brain. Machine learning is also known as predictive analytics when it is used to solve business challenges.

There are three types of machine learning algorithms in the field of machine learning:

- **Supervised Learning**: These algorithms have a goal or a known result variable that the supplied collection of predictors will predict. We are able to use supervised learning to create functions that connect our desired outputs to our inputs
- **Unsupervised Learning**: In this form of method, the aim or result is not known. There isn't a variable to forecast. Clustering techniques of this type are extensively employed. A collection of data in several groups
- **Reinforcement Learning**: When a machine learns by itself, it is called reinforcement learning. Using trial and error in the given surroundings. The machine also improves on previous models. Learning outcomes will allow you to make accurate judgments in a particular situation.

7.2 Deep Learning

Deep learning (also known as deep structured learning) is a type of machine learning technology that uses artificial neural networks to learn representations. There are three types of learning: supervised, semi-supervised, and unsupervised.

Deep-learning architectures such as deep neural networks, deep belief networks, deep reinforcement learning, recurrent neural networks, and convolutional neural networks have been used in fields such as computer vision, speech recognition, natural language processing, machine translation, bioinformatics, drug design, medical image analysis, material inspection, and board game programs, with results comparable to, and in some cases exceeding, human performance.

Information processing and dispersed communication nodes in biological systems inspired artificial neural networks (ANNs). ANNs differ from biological brains in a number of ways. Artificial neural networks, in particular, are static and symbolic, whereas most living animals' biological brains are dynamic (plastic) and analogue.

- **Artificial neural network (ANN)**

An artificial neural network (ANN) having numerous layers between the input and output layers is known as a deep neural network (DNN). Neural networks come in a variety of shapes and sizes, but they all include the same basic components: neurons, synapses, weights, biases, and functions. These components work in the same way as human brains and can be trained just like any other machine learning algorithm. [23][24]

- **Recurrent neural networks (RNNs)**

Recurrent neural networks (RNNs) are employed in applications like language modelling because input can flow in either direction. For this purpose, long short-term memory is very useful. [25][26][27][28]

- **Convolutional deep neural networks (CNNs)**

In computer vision, convolutional deep neural networks (CNNs) are used. Acoustic modelling for automatic speech recognition has also used CNNs (ASR). [30]

8. Machine Learning Algorithms Used In This Project

We chose multi-dimensional methods that map to our dataset from a list of several machine learning algorithms accessible. Keeping the project's goal in mind, we picked the following algorithms for the best and most accurate results:

1 Decision Trees
2 k-Nearest Neighbor
3 MLP
4 Random Forest
5 Convolutional Neural Network

Through the above-mentioned machine learning algorithms, the table comprising all of the characteristics of the segmented images will be utilized to train the Machine Learning Neural Network. [2][3]

8.1 Decision Trees

A Decision Trees are a graphical depiction of multiple decisions and their potential outcomes, such as events, resource costs, and utility. It is a decision aid that employs a tree-like graph or decision model. A decision tree is made up of three parts:

- Internal node: Represents an attribute test.
- Branch: Represents the test's result.
- Leaf Node: indicates a specific class label, i.e. the decision taken after all of the characteristics have been computed.

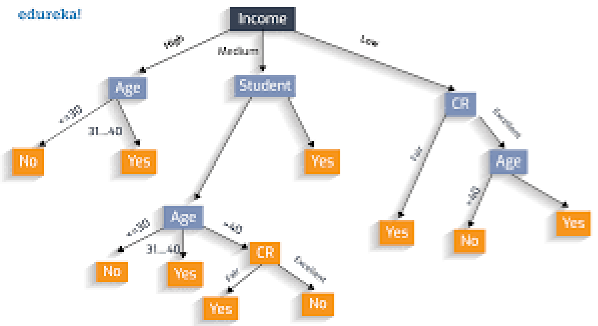


Fig: 8.1

8.1.1 Types Of Decision Trees:

Decision trees may be divided into two types:

- Classification Trees: When the response variable is categorical, they are utilized. Based on the response variables, datasets are divided into several classifications.
- Regression Trees: The response variable might be either continuous or numerical in nature. Binary variable regression trees and continuous variable decision trees are two types of binary variable regression trees.

8.1.1.1 Benefits of Using Decision Trees

1 The visual depiction makes it simple to comprehend and communicate to everyone.

2 You may utilize both category and numerical data.

3 When huge datasets are employed, it performs well.

4 It may be utilized in situations where variables are related in a linear or non-linear way.

5 They are unaffected by missing values and outlines, which saves time while exploring data.

6 Using forward and backward approaches, the best conclusion may be selected.

8.1.1.2 Decision Trees Have A Number Of Drawbacks

1. Decision trees are less precise than other methods.
2. When continuous data is employed, they might cause instability.
3. Large decision trees with several branches can become complicated and difficult to comprehend.
4. They are based on expectations and real facts, and actual results may differ

8.2 K-nearest neighbors (KNN)

K-nearest neighbors (KNN) algorithm is a type of supervised ML algorithm which can be used for both classification as well as regression predictive problems. However, it is mainly used for classification predictive problems in industry. [7][8]

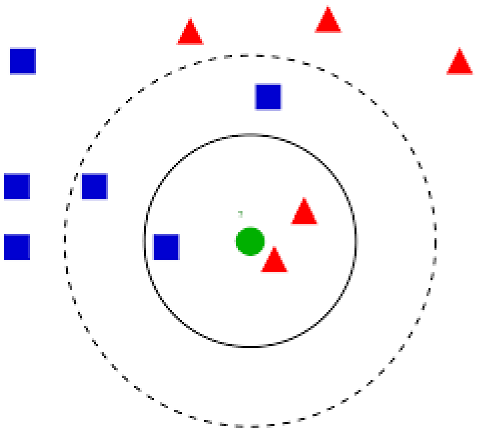


Fig: 8.2

8.2.1 Advantages of KNN

1. It's easy to put into practice and comprehend.
2. When used for classification and regression, it can learn non-linear decision boundaries. Can devised a highly adaptable decision boundary by varying the value of K.
3. There has been no training. Classification/regression time: There is no explicit training stage in the KNN algorithm, and all of the work is done during prediction.
4. Constantly develops in response to fresh data: Because there is no explicit training stage, the prediction is modified when fresh data is added to the dataset without having to retrain a new model.
5. Hyper parameters with a single value: The value of K is the only hyper parameter. This facilitates hyper parameter adjustment.

8.2.2 **Disadvantages of KNN**

1 High prediction complexity for large datasets: Not ideal for large datasets, as each prediction requires the processing of the full training data. Each prediction has a time complexity of O (MN log (k)), where M is the data dimension and N is the amount or number of occurrences in the training data. To overcome this issue and make KNN quicker, there are specific techniques of structuring data.

2 With more dimensions, there is a higher level of prediction complexity. For greater dimensional data, the prediction complexity in supervised learning increases (see the dependence of time complexity from the previous point on the dimension d).

3 Optional distance metric: There are several distance measurements from which to choose. Euclidean, Manhattan, Murkowski, hamming distance, and other distance metrics are commonly utilized.

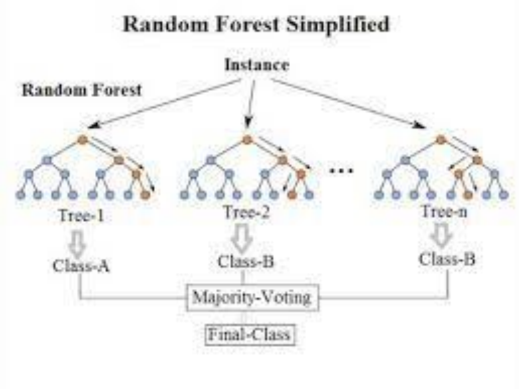
4 KNN assigns equal weight to all features: Because KNN expects points to be near in ALL dimensions, it may overlook points that are extremely close in some dimensions but far apart in others. This can be changed by selecting a suitable distance measure. Furthermore, if distinct characteristics have different ranges, this shows it is sensitive. This can be handled by scaling features with proper pre-processing.

5 Excessive sensitivity to outliers: A single incorrectly classified example might cause the class borders to shift. Because the average separation tends to be higher for higher dimensions (curse of dimensionality), outliers can have a greater influence on outliers in one dimension

.

8.3 Random Forest

Random forest machine learning algorithm that produces, even without hyper-parameter tuning, a great result most of the time. Because of its simplicity and versatility, it is also one of the most often used algorithms (it can be used for both classification and regression tasks). We'll look at how the random forest method works, how it varies from other algorithms, and how to use it in this post. [6][8]



8.3.1 Advantages of Random Forest

1. It helps to enhance decision tree accuracy by reducing over fitting.
2. It can handle both classification and regression issues with ease.
3. It can handle both category and continuous data.
4. It automates the process of filling in missing values in data.
5. Because it employs a rule-based approach, no data normalization is necessary

8.3.2 Disadvantages of Random Forest

1. It needs a significant amount of computer power as well as resources because it constructs several trees and combines their outcomes.
2. It also takes a long time to train since it uses a number of decision trees to select the class.
3. It also lacks interpretability due to the ensemble of decision trees and fails to evaluate the relevance of each variable
4. It also lacks interpretability due to the ensemble of decision trees and fails to evaluate the relevance of each variable

8.4 Convolutional neural network (CNN)

Convolutional neural network (CNN), a class of artificial neural networks that has become dominant in various computer vision tasks, is attracting interest across a variety of domains, including radiology. CNN uses several building blocks like as convolution layers, pooling layers, and fully connected layers to learn spatial hierarchies of information automatically and adaptively through back propagation. CNN is a classification framework that divides pictures into labelled categories. The CNN's various layers extract image features before learning to classify the images. As a result, a typical CNN's outputs indicate the classes or labels of the classes that the CNN has learned to categorize. [30][31]

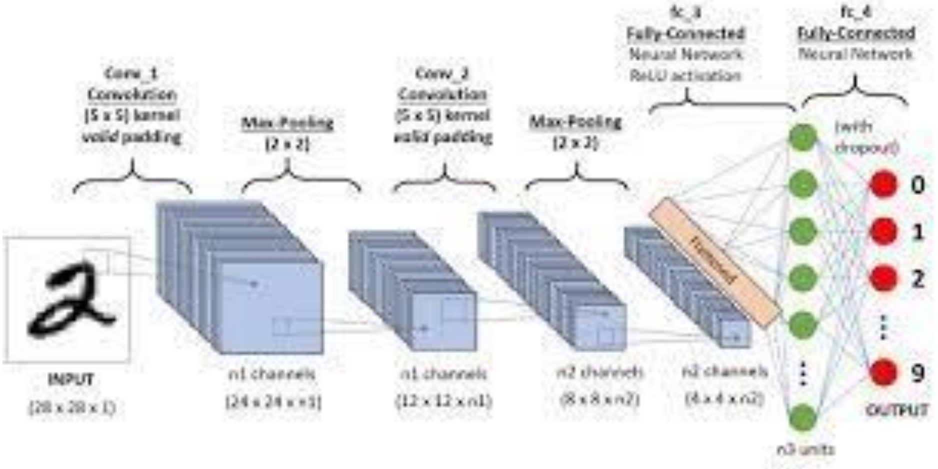


Fig: 8.4

8.4.1 Advantages of Convolutional neural network

1. Features are automatically inferred and modified to get the desired result. It is not necessary to extract features ahead of time. This eliminates the need for time-consuming machine learning approaches.
2. Automatically trained robustness to natural fluctuations in the data.
3. Many different applications and data kinds can benefit from the same neural network-based technique.
4. GPUs can execute massively parallel computations that are scalable for vast amounts of data. Furthermore, it provides superior performance outcomes while dealing with large amounts of data.
5. The deep learning architecture is adaptable, which means it may be used to solve new challenges in the future

8.4.2 Disadvantages of Convolutional neural network

1. To perform better than other strategies, it needs a big volume of data.
2. Because of the complicated data models, training is exceedingly costly. Deep learning also necessitates the use of pricey GPUs and hundreds of workstations. The users' costs will rise as a result of this.
3. Because it necessitates knowledge of topology, training technique, and other characteristics, there is no standard theory to aid you in choosing the correct deep

learning tools. As a result, it is difficult for less competent individuals to embrace it.

4. It is difficult to grasp output based just on learning, and therefore necessitates the use of classifiers. Such tasks are carried out using algorithms based on convolutional neural networks.

8.5 Multi-Layer Perceptron (MLP)

The multi-layer perceptron (MLP) is a feed forward neural network augmentation. It has three layers: an input layer, an output layer, and a hidden layer. The input signal to be processed is received by the input layer. The output layer is responsible for tasks such as prediction and categorization. The real computational engine of the MLP is an arbitrary number of hidden layers inserted between the input and output layers. In an MLP, data travels from input to output layer in the forward direction, similar to a feed forward network. The back propagation learning technique is used to train the neurons in the MLP. MLPs can tackle issues that aren't linearly separable and are meant to approximate any continuous function. Pattern categorization, identification, prediction, and approximation are some of MLP's most common applications. [8][9]

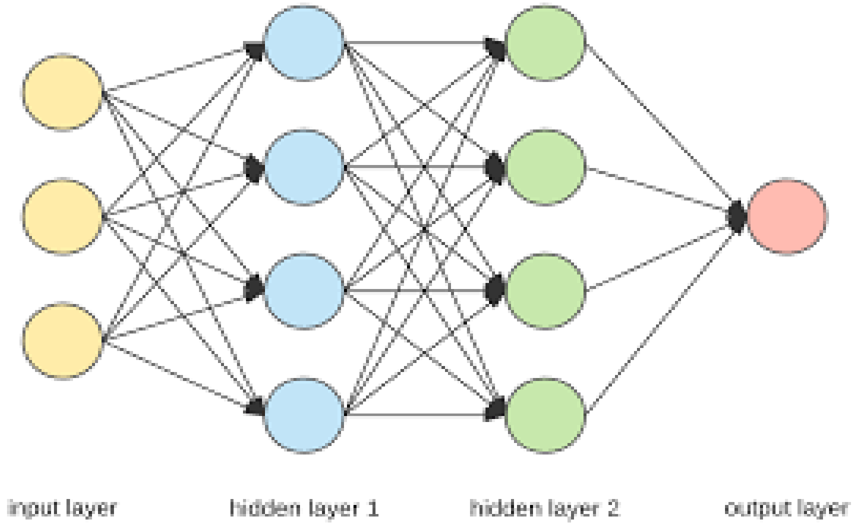


Fig: 8.5

8.5.1 Advantages of Multi-Layer Perceptron

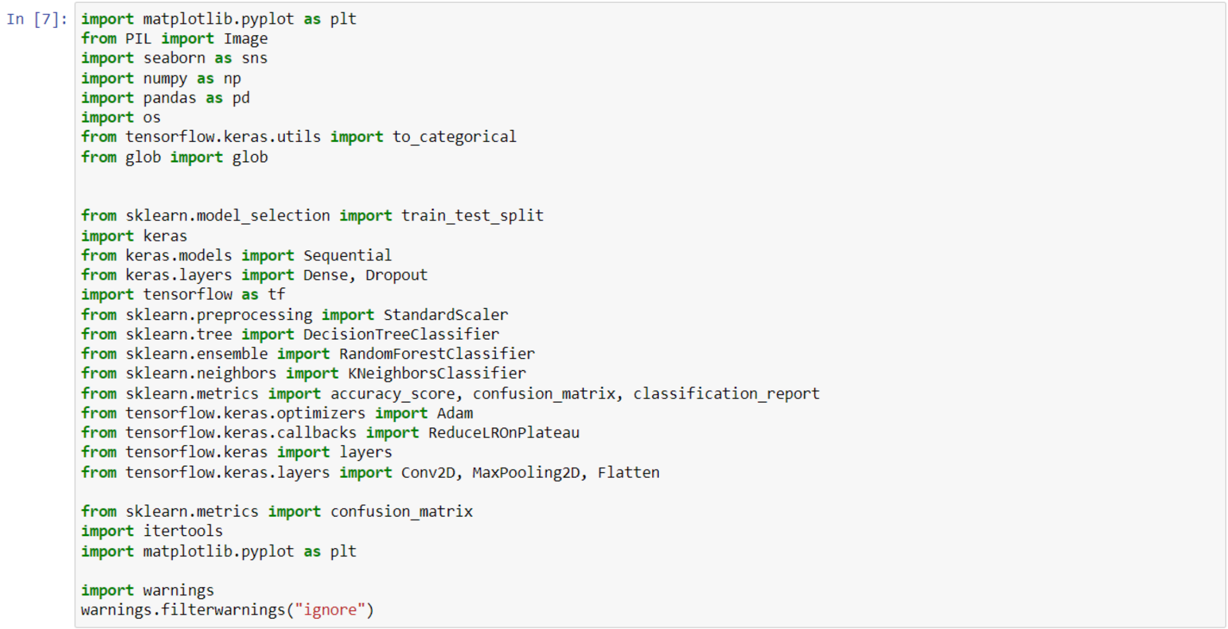
1. Can be applied to complex non-linear problems.
2. Works well with large input data.
3. Provides quick predictions after training.
4. The same accuracy ratio can be achieved even with smaller data.

8.5.2 Disadvantages of Multi-Layer Perceptron

1. It is not known to what extent each independent variable is affected by the dependent variable. Computations are difficult and time consuming.
2. The proper functioning of the model depends on the quality of the training

9. Code Snips

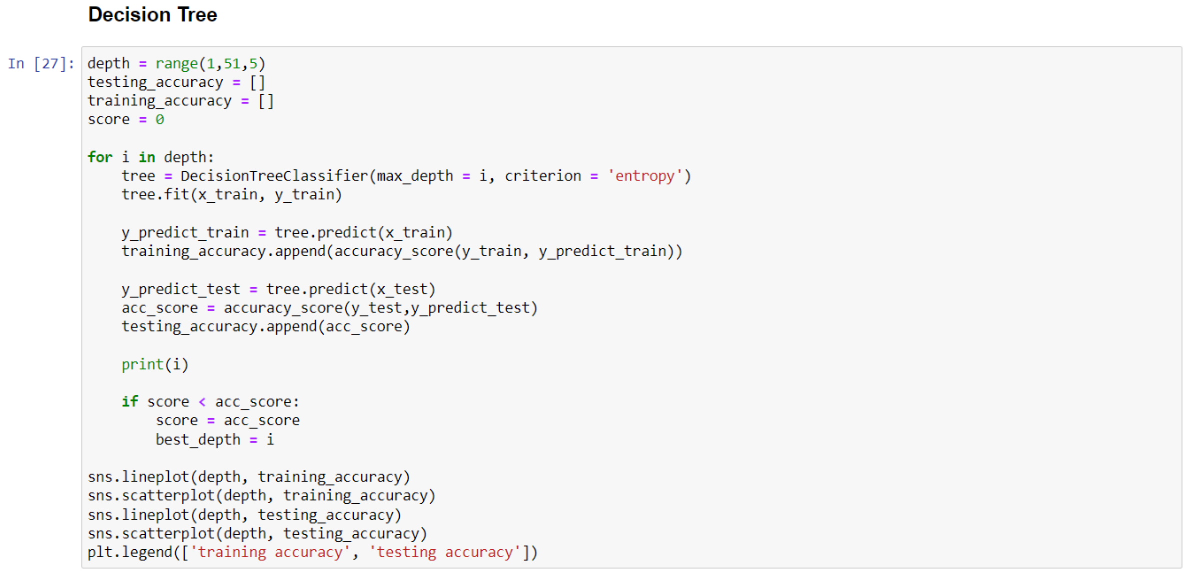
9.1 Libraries



In [7]:

```
import matplotlib.pyplot as plt
from PIL import Image
import seaborn as sns
import numpy as np
import pandas as pd
import os
from tensorflow.keras.utils import to_categorical
from glob import glob


from sklearn.model_selection import train_test_split
import keras
from keras.models import Sequential
from keras.layers import Dense, Dropout
import tensorflow as tf
from sklearn.preprocessing import StandardScaler
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.callbacks import ReduceLROnPlateau
from tensorflow.keras import layers
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten

from sklearn.metrics import confusion_matrix
import itertools
import matplotlib.pyplot as plt

import warnings
warnings.filterwarnings("ignore")
```

9.2 Decision Tree

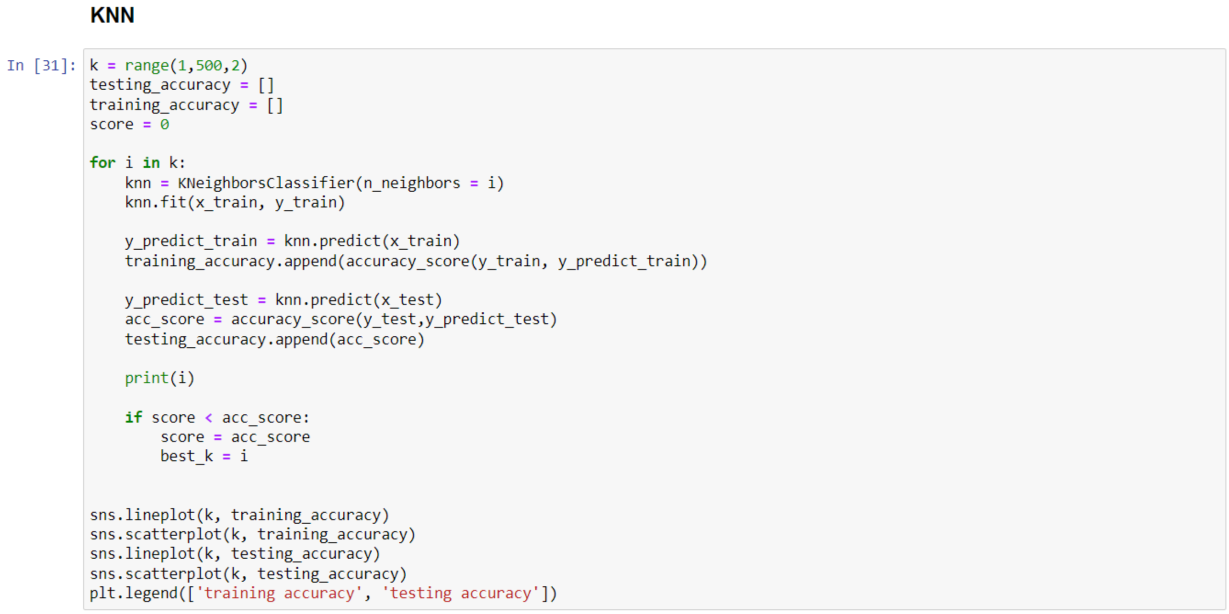


Decision Tree

In [27]:

```
depth = range(1,51,5)
testing_accuracy = []
training_accuracy = []
score = 0

for i in depth:
    tree = DecisionTreeClassifier(max_depth = i, criterion = 'entropy')
    tree.fit(x_train, y_train)

    y_predict_train = tree.predict(x_train)
    training_accuracy.append(accuracy_score(y_train, y_predict_train))

    y_predict_test = tree.predict(x_test)
    acc_score = accuracy_score(y_test,y_predict_test)
    testing_accuracy.append(acc_score)

    print(i)

    if score < acc_score:
        score = acc_score
        best_depth = i

sns.lineplot(depth, training_accuracy)
sns.scatterplot(depth, training_accuracy)
sns.lineplot(depth, testing_accuracy)
sns.scatterplot(depth, testing_accuracy)
plt.legend(['training accuracy', 'testing accuracy'])
```

9.3 KNN

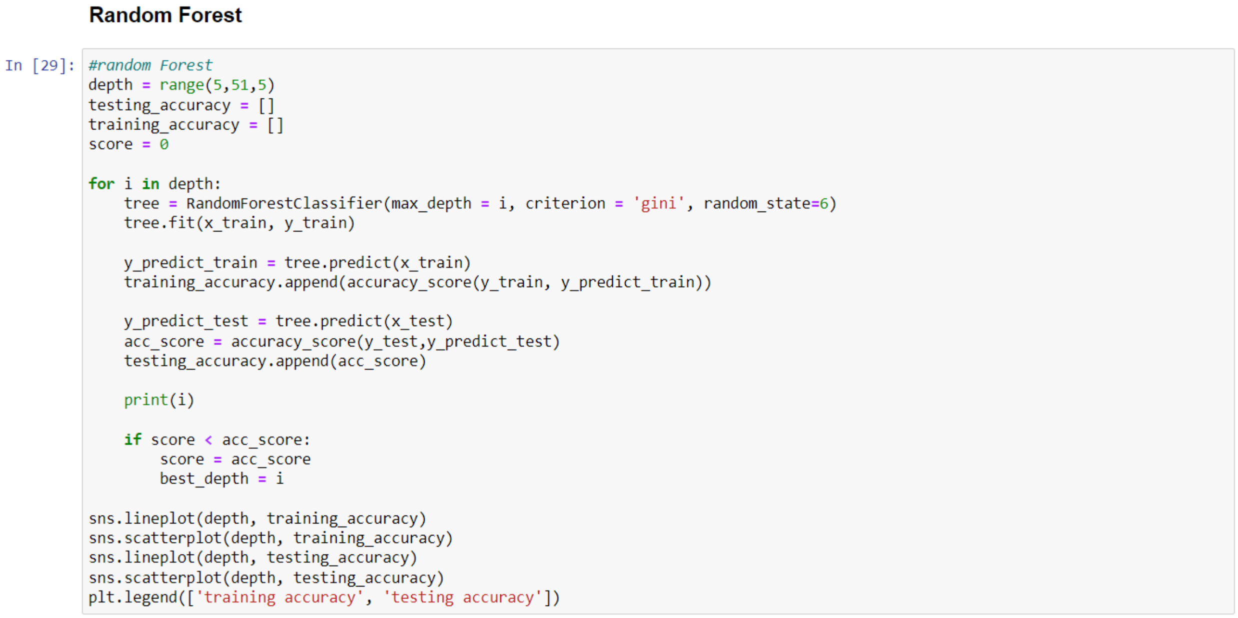


KNN

In [31]:

```
k = range(1,500,2)
testing_accuracy = []
training_accuracy = []
score = 0

for i in k:
    knn = KNeighborsClassifier(n_neighbors = i)
    knn.fit(x_train, y_train)

    y_predict_train = knn.predict(x_train)
    training_accuracy.append(accuracy_score(y_train, y_predict_train))

    y_predict_test = knn.predict(x_test)
    acc_score = accuracy_score(y_test,y_predict_test)
    testing_accuracy.append(acc_score)

    print(i)

    if score < acc_score:
        score = acc_score
        best_k = i


sns.lineplot(k, training_accuracy)
sns.scatterplot(k, training_accuracy)
sns.lineplot(k, testing_accuracy)
sns.scatterplot(k, testing_accuracy)
plt.legend(['training accuracy', 'testing accuracy'])
```

9.4 Random Forest

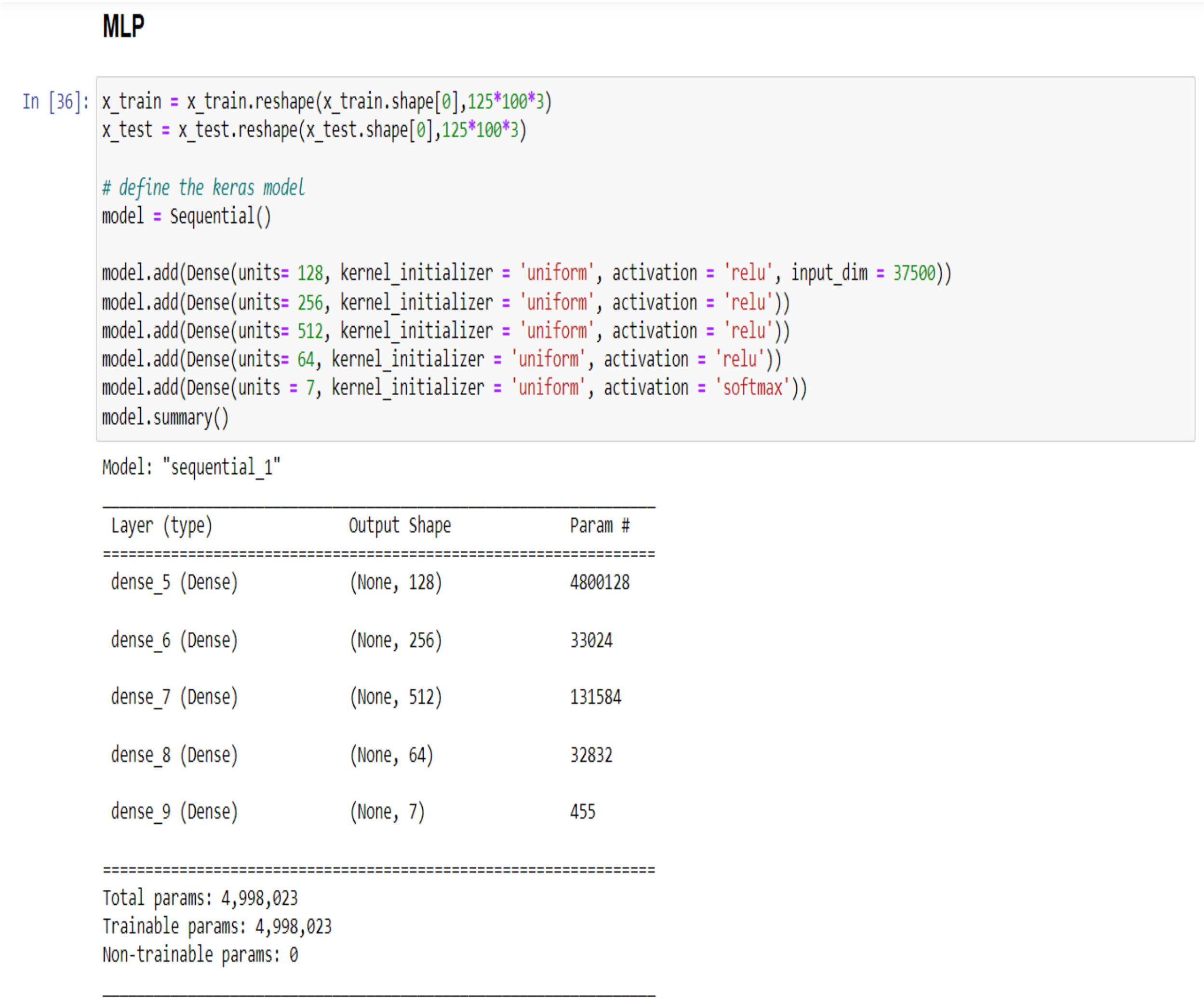


Random Forest

In [29]:

```
#random Forest
depth = range(5,51,5)
testing_accuracy = []
training_accuracy = []
score = 0

for i in depth:
    tree = RandomForestClassifier(max_depth = i, criterion = 'gini', random_state=6)
    tree.fit(x_train, y_train)

    y_predict_train = tree.predict(x_train)
    training_accuracy.append(accuracy_score(y_train, y_predict_train))

    y_predict_test = tree.predict(x_test)
    acc_score = accuracy_score(y_test,y_predict_test)
    testing_accuracy.append(acc_score)

    print(i)

    if score < acc_score:
        score = acc_score
        best_depth = i

sns.lineplot(depth, training_accuracy)
sns.scatterplot(depth, training_accuracy)
sns.lineplot(depth, testing_accuracy)
sns.scatterplot(depth, testing_accuracy)
plt.legend(['training accuracy', 'testing accuracy'])
```

9.5 ANN (MLP)

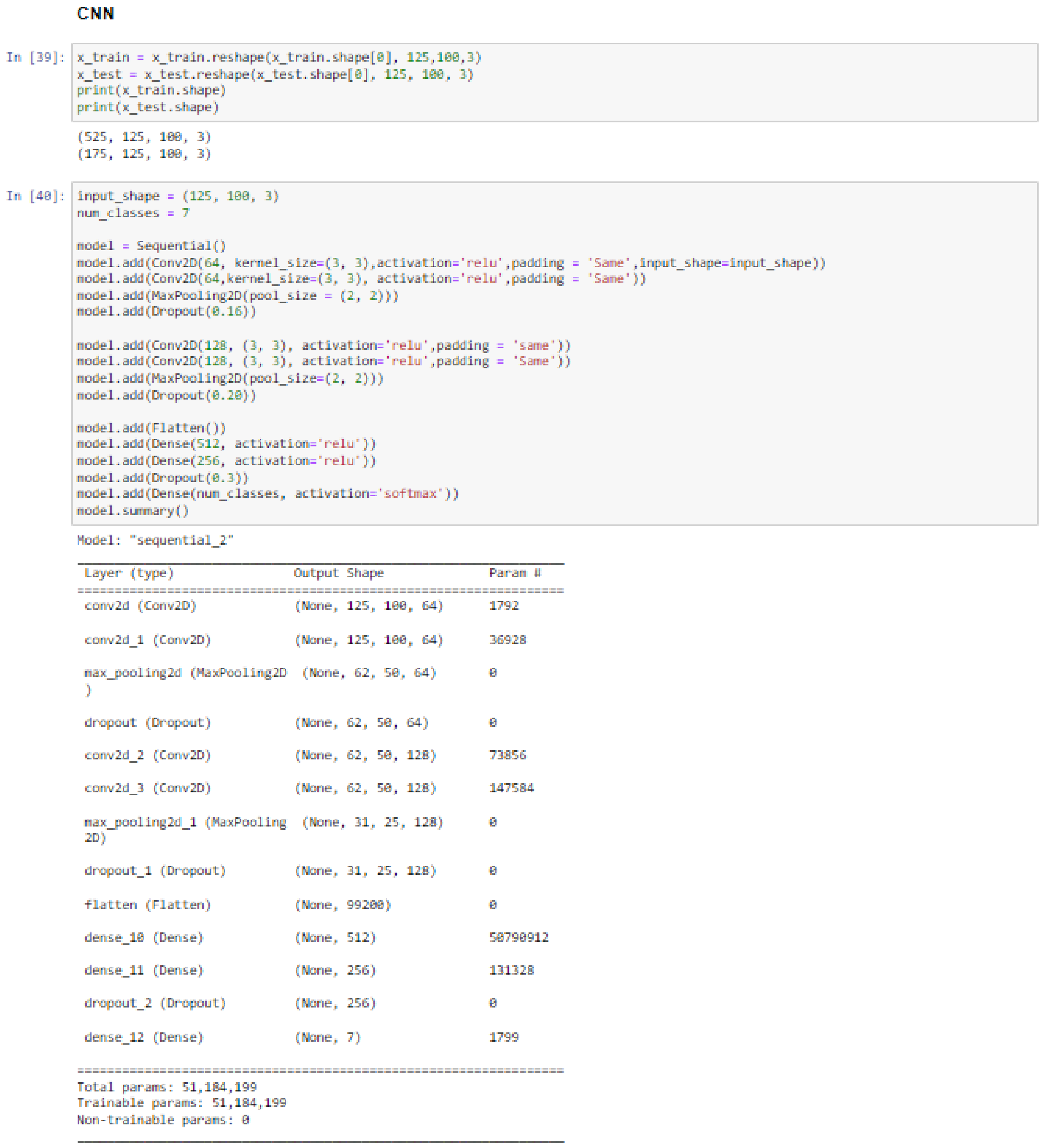


MLP

```
In [36]: x_train = x_train.reshape(x_train.shape[0],125*100*3)
         x_test = x_test.reshape(x_test.shape[0],125*100*3)

         # define the keras model
         model = Sequential()

         model.add(Dense(units= 128, kernel_initializer = 'uniform', activation = 'relu', input_dim = 37500))
         model.add(Dense(units= 256, kernel_initializer = 'uniform', activation = 'relu'))
         model.add(Dense(units= 512, kernel_initializer = 'uniform', activation = 'relu'))
         model.add(Dense(units= 64, kernel_initializer = 'uniform', activation = 'relu'))
         model.add(Dense(units = 7, kernel_initializer = 'uniform', activation = 'softmax'))
         model.summary()
```

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense_5 (Dense)             (None, 128)               4800128

 dense_6 (Dense)             (None, 256)               33024

 dense_7 (Dense)             (None, 512)               131584

 dense_8 (Dense)             (None, 64)                32832

 dense_9 (Dense)             (None, 7)                 455

=================================================================
Total params: 4,998,023
Trainable params: 4,998,023
Non-trainable params: 0
_________________________________________________________________
```

9.6 CNN



CNN

```
In [39]: x_train = x_train.reshape(x_train.shape[0], 125,100,3)
         x_test = x_test.reshape(x_test.shape[0], 125, 100, 3)
         print(x_train.shape)
         print(x_test.shape)

         (525, 125, 100, 3)
         (175, 125, 100, 3)
```

```
In [40]: input_shape = (125, 100, 3)
         num_classes = 7

         model = Sequential()
         model.add(Conv2D(64, kernel_size=(3, 3),activation='relu',padding = 'Same',input_shape=input_shape))
         model.add(Conv2D(64,kernel_size=(3, 3), activation='relu',padding = 'Same'))
         model.add(MaxPooling2D(pool_size = (2, 2)))
         model.add(Dropout(0.16))

         model.add(Conv2D(128, (3, 3), activation='relu',padding = 'same'))
         model.add(Conv2D(128, (3, 3), activation='relu',padding = 'Same'))
         model.add(MaxPooling2D(pool_size=(2, 2)))
         model.add(Dropout(0.20))

         model.add(Flatten())
         model.add(Dense(512, activation='relu'))
         model.add(Dense(256, activation='relu'))
         model.add(Dropout(0.3))
         model.add(Dense(num_classes, activation='softmax'))
         model.summary()
```

```
Model: "sequential_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 125, 100, 64)      1792

 conv2d_1 (Conv2D)           (None, 125, 100, 64)      36928

 max_pooling2d (MaxPooling2D  (None, 62, 50, 64)       0
 )

 dropout (Dropout)           (None, 62, 50, 64)        0

 conv2d_2 (Conv2D)           (None, 62, 50, 128)       73856

 conv2d_3 (Conv2D)           (None, 62, 50, 128)       147584

 max_pooling2d_1 (MaxPooling  (None, 31, 25, 128)      0
 2D)

 dropout_1 (Dropout)         (None, 31, 25, 128)       0

 flatten (Flatten)           (None, 99200)             0

 dense_10 (Dense)            (None, 512)               50790912

 dense_11 (Dense)            (None, 256)               131328

 dropout_2 (Dropout)         (None, 256)               0

 dense_12 (Dense)            (None, 7)                 1799

=================================================================
Total params: 51,184,199
Trainable params: 51,184,199
Non-trainable params: 0
_________________________________________________________________
```

10. Preprocessing of the data

We found out all the null values available in the dataset or if that cell was incomplete then we replaced that cell value with 0. Then later on still we find out that age column was causing ambiguity that’s why we dropped the age from feature table. Then furthermore to reduce more ambiguity we dropped the image id column we were using the ham if to classify and group same cancers. As it was much easier to group the different datasets on the basis of cancer id or HAM id no given to them. We used the HAM ID to compare and identify the same cancer using HAM ID to make sure the algorithm runs appropriately.

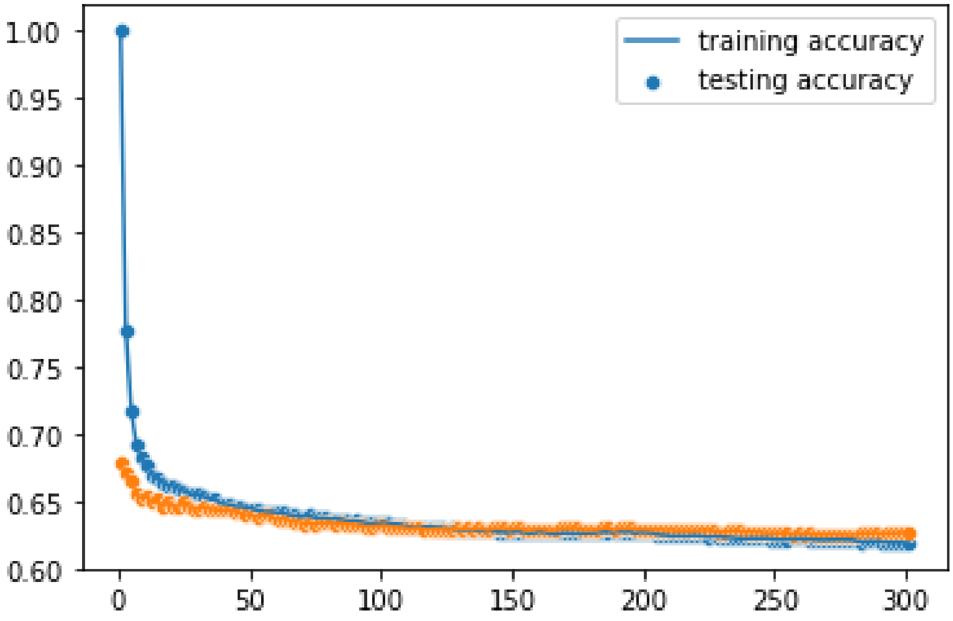
In initial we picked 700 picture and dataset to find out that whether our data is working appropriately or not 100 samples of each dataset. Furthermore in 2nd phase we used the original data set but there was one problem with this that cancer type Melanocytic Nevi (NV) had more data samples compared to its other predecessors therefore we devised a way to multiply dataset with appropriate numbers to make it equal with the surplus of the specified dataset samples.

11. Results

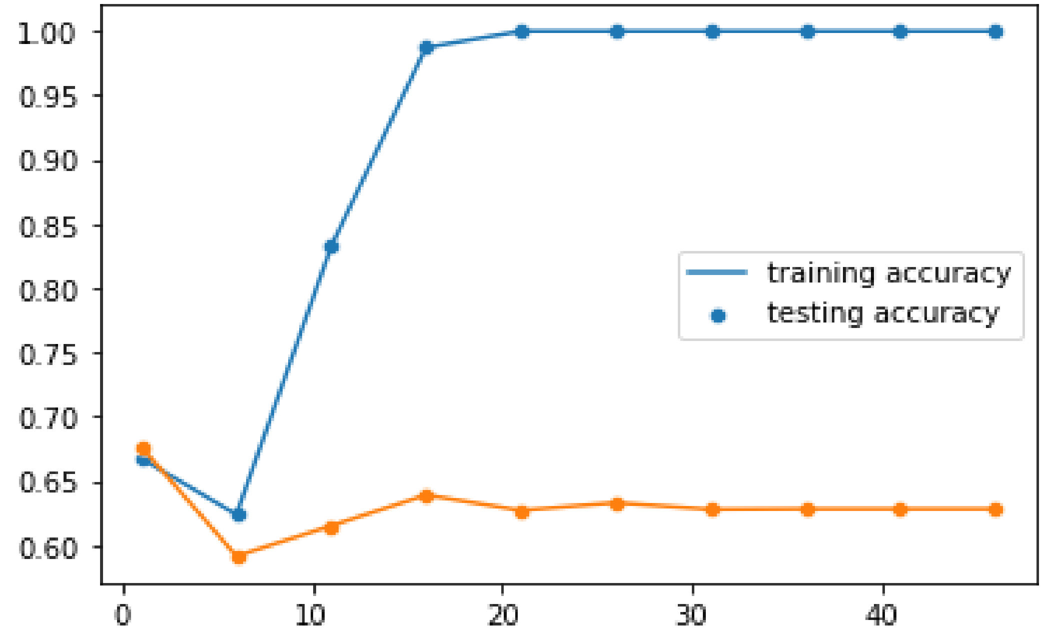
11.1 Training Model

1. Graphical representation of result from Original Data

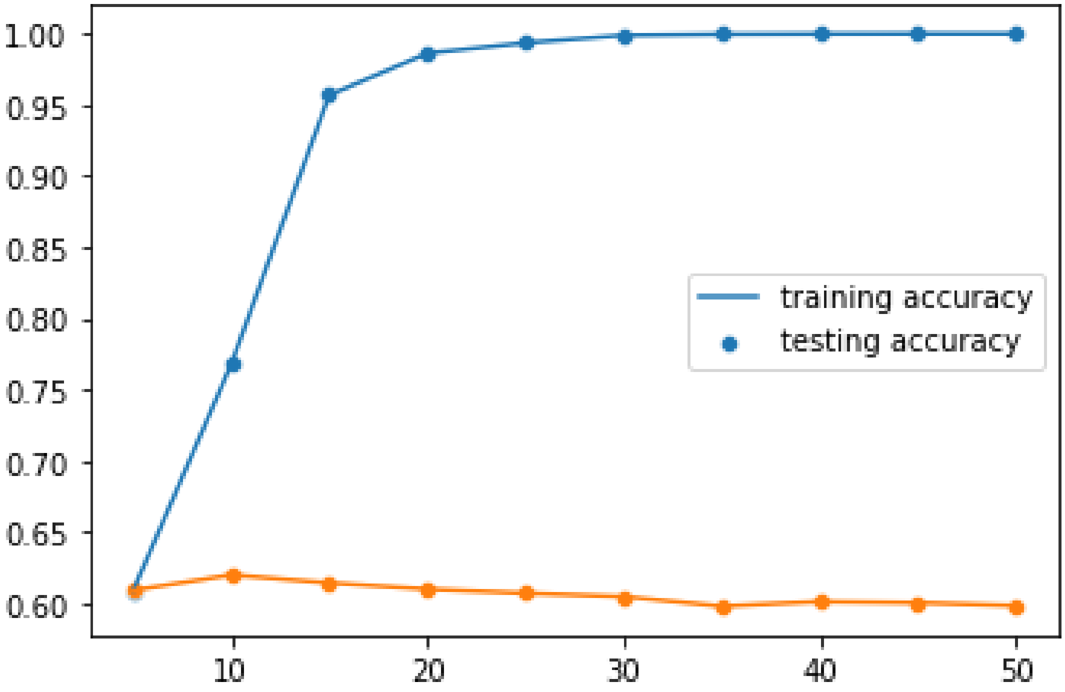
   - KNN



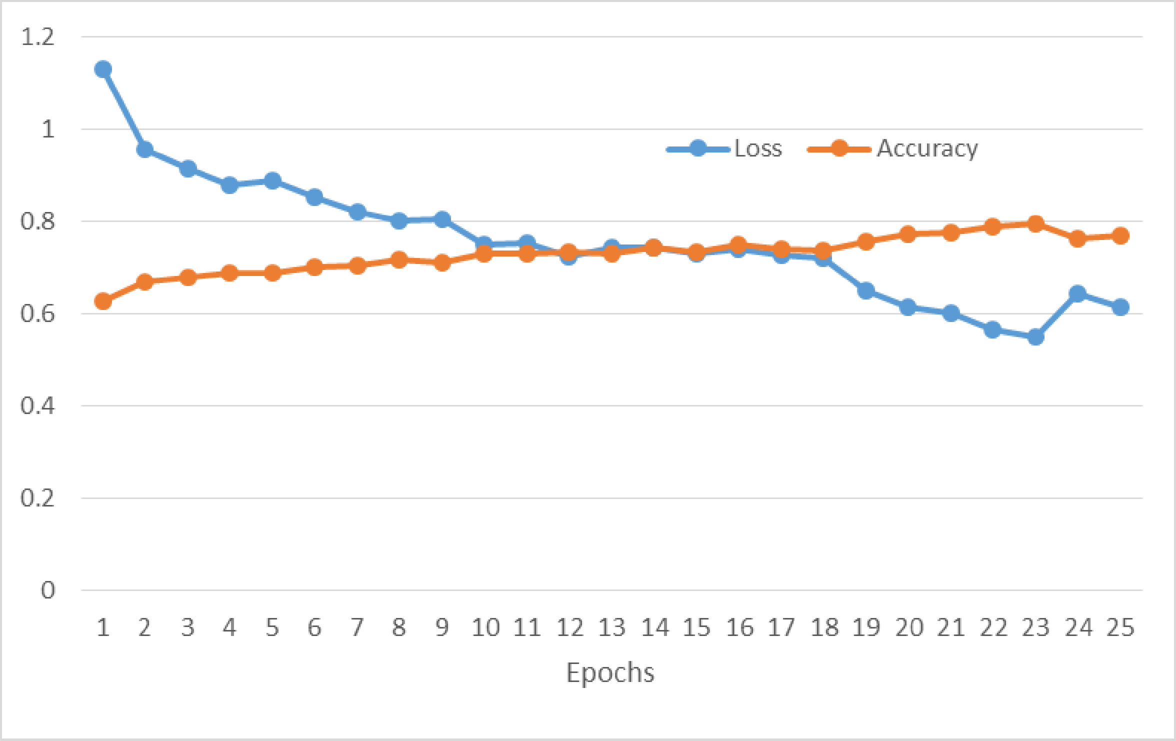
- Decision Tree



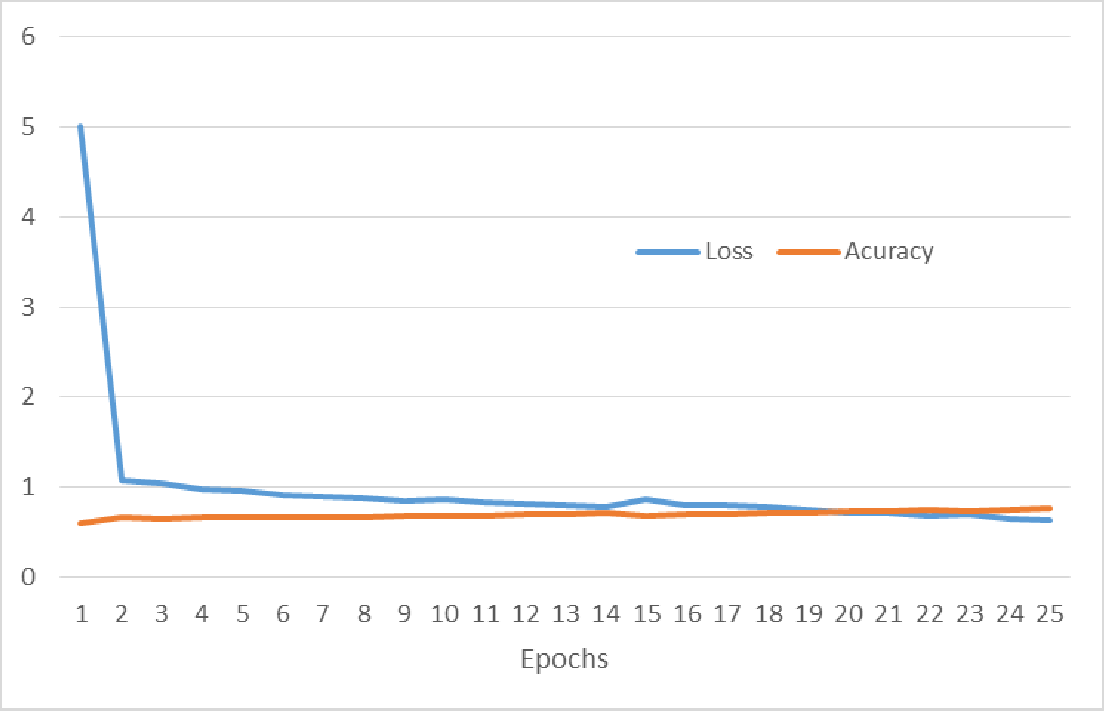
- Random Tree Forest



- MLP

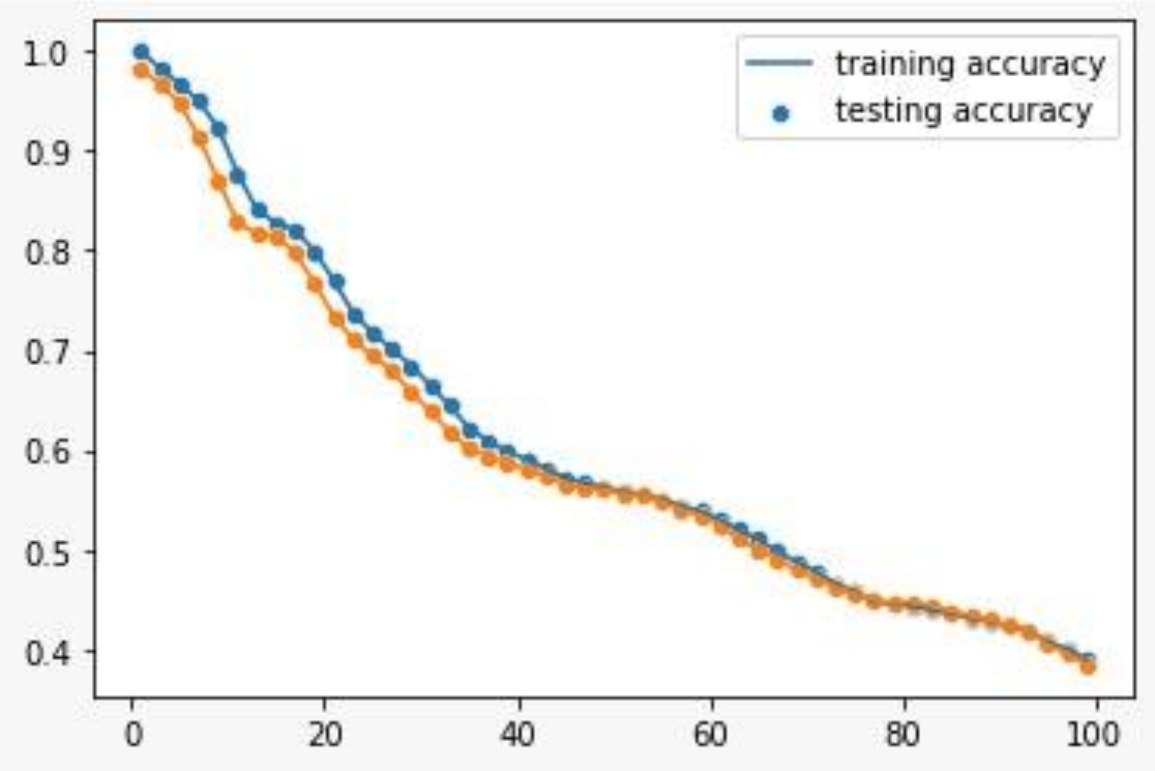

1.2
1
0.8
0.6
0.4
0.2
0
Loss
Accuracy
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
Epochs

- CNN

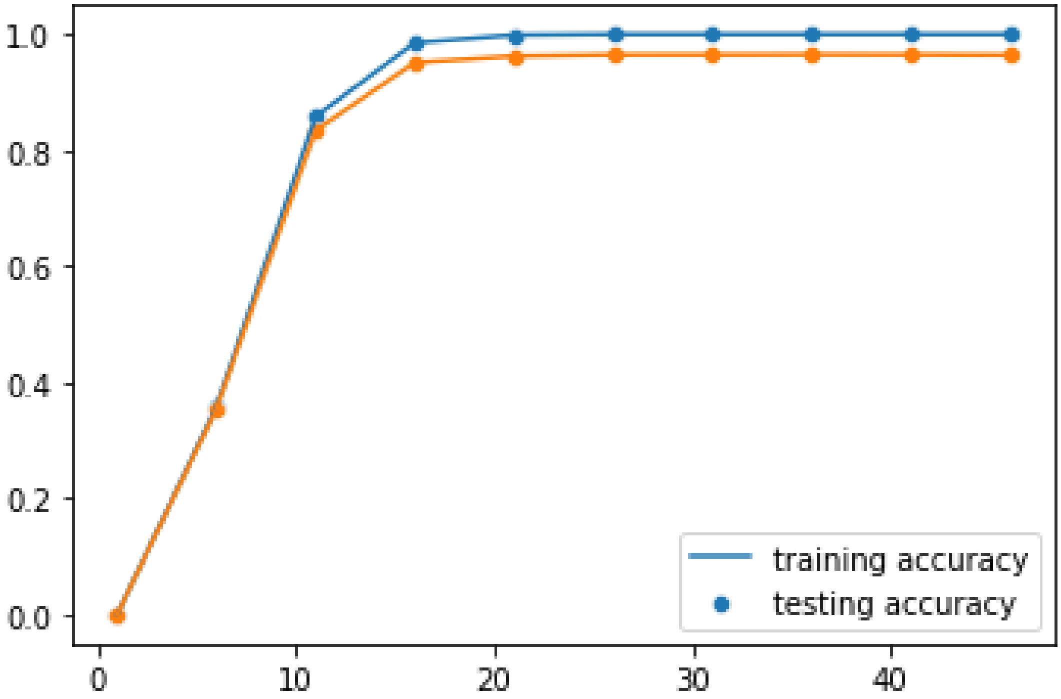

6
5
4
3
2
1
0
Loss
Acuracy
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
Epochs

2. Graphical representation of result from Augmented Data (Increased)

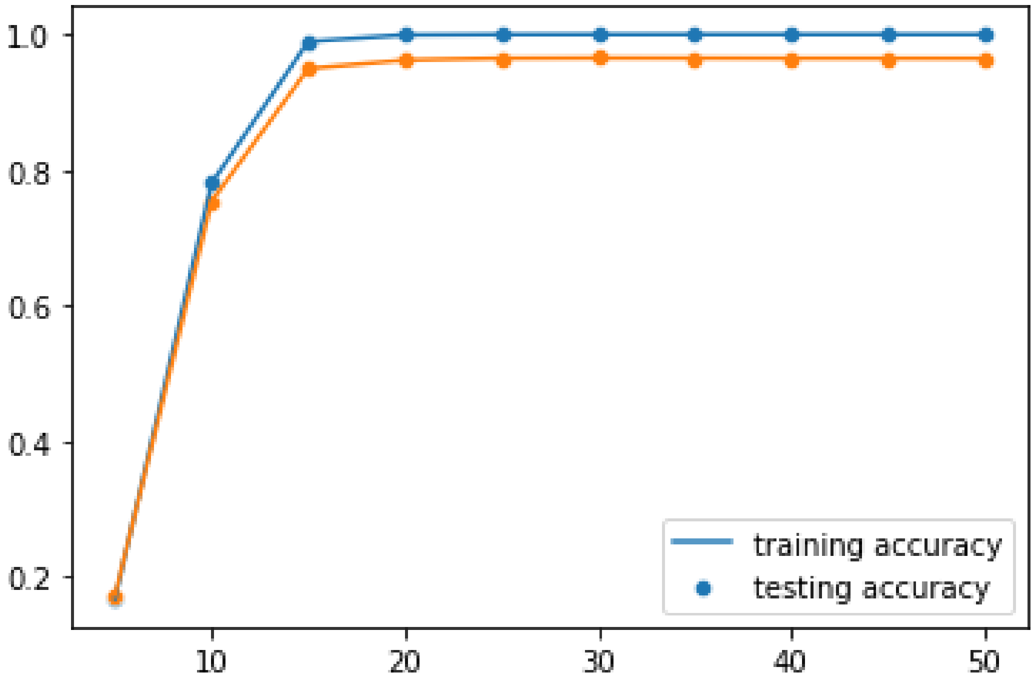
- KNN



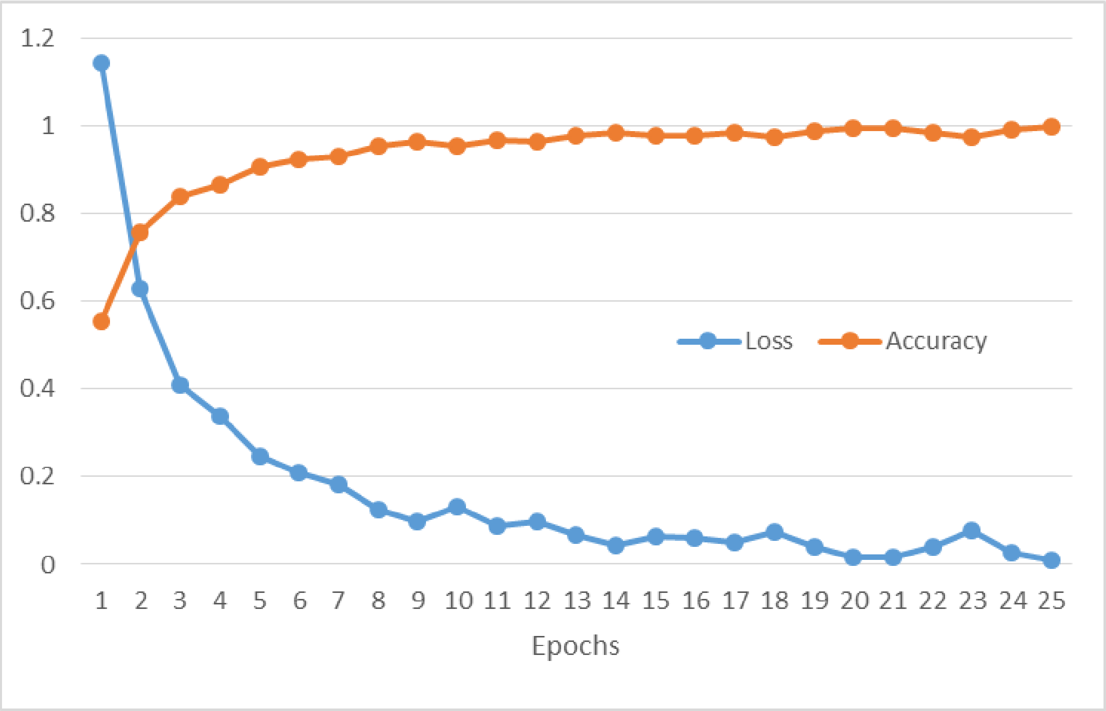
- Decision Tree



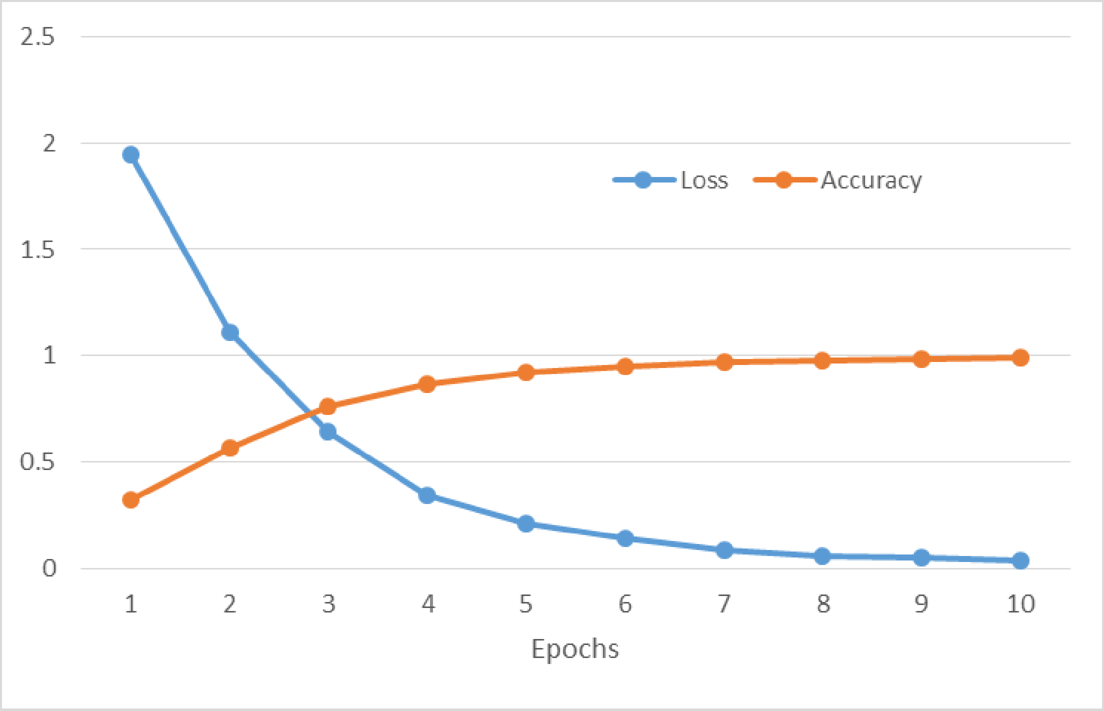
- Random Tree Forest



- MLP

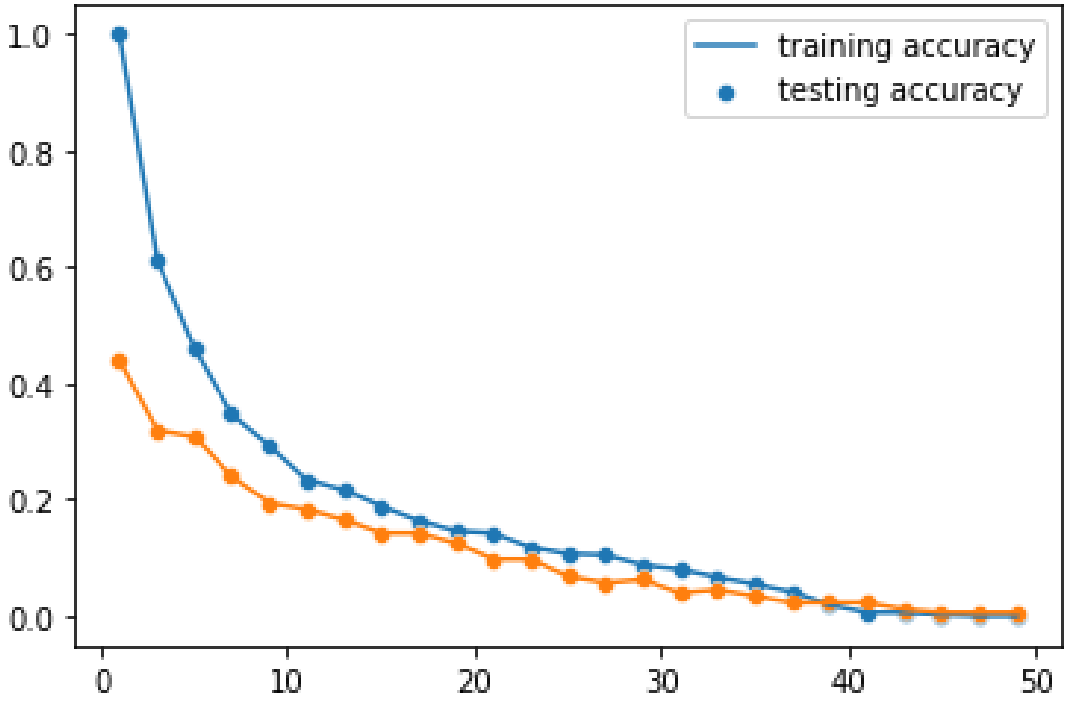


- CNN

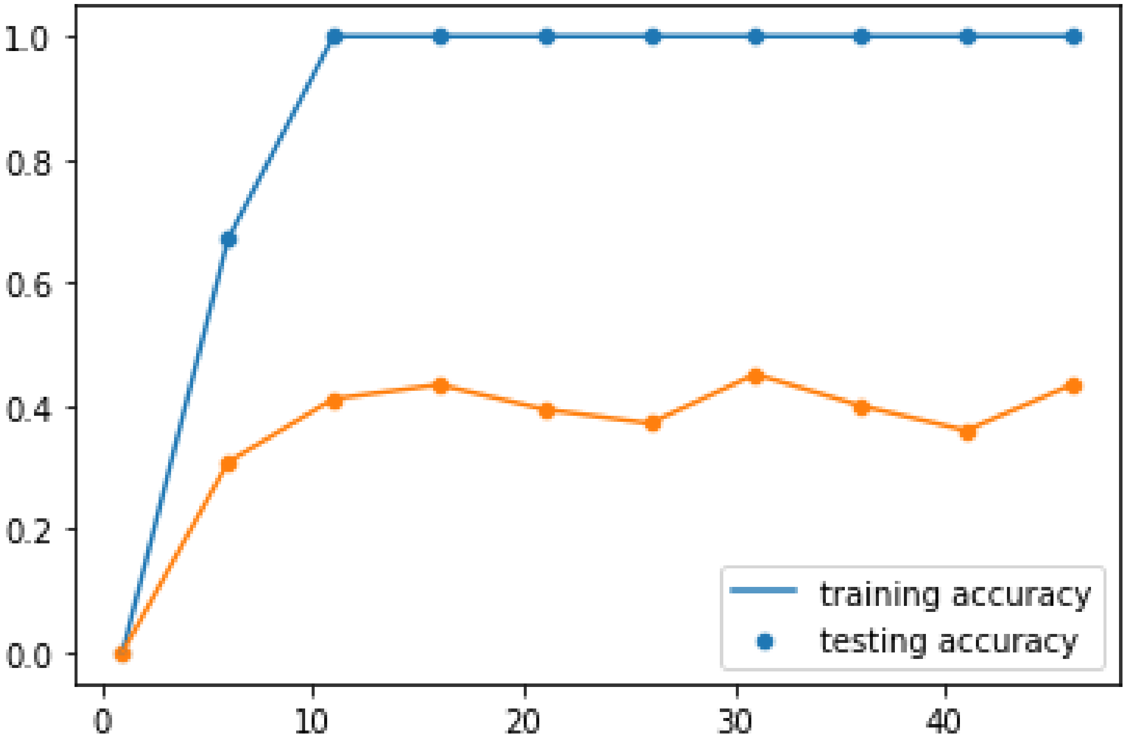


3. Graphical representation of result from Augmented Data (Decreased)

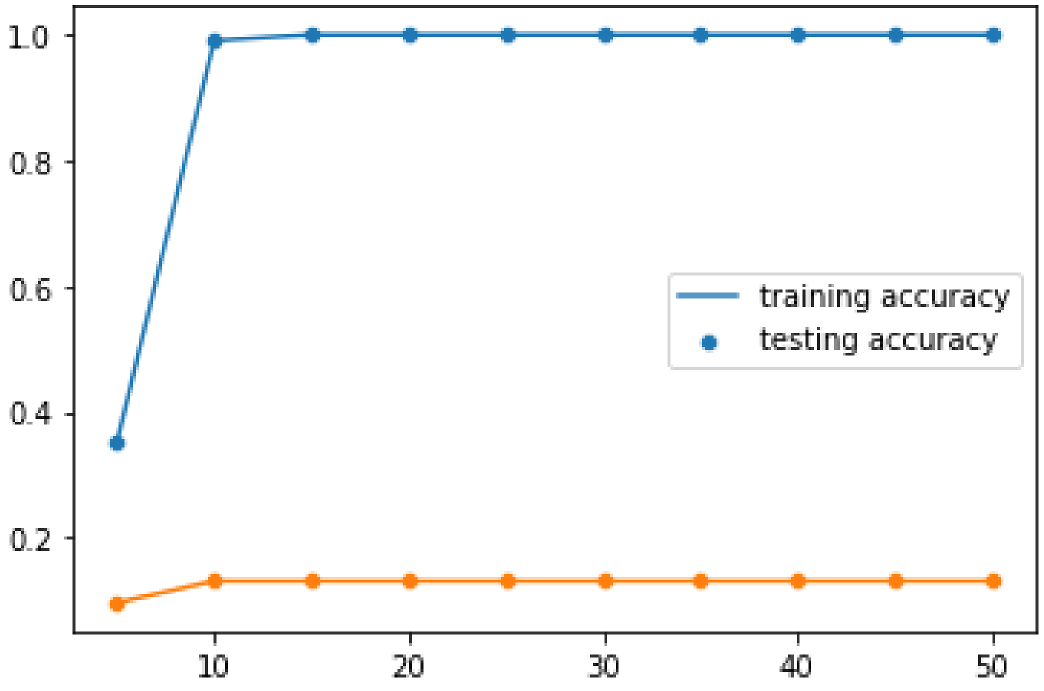
- KNN



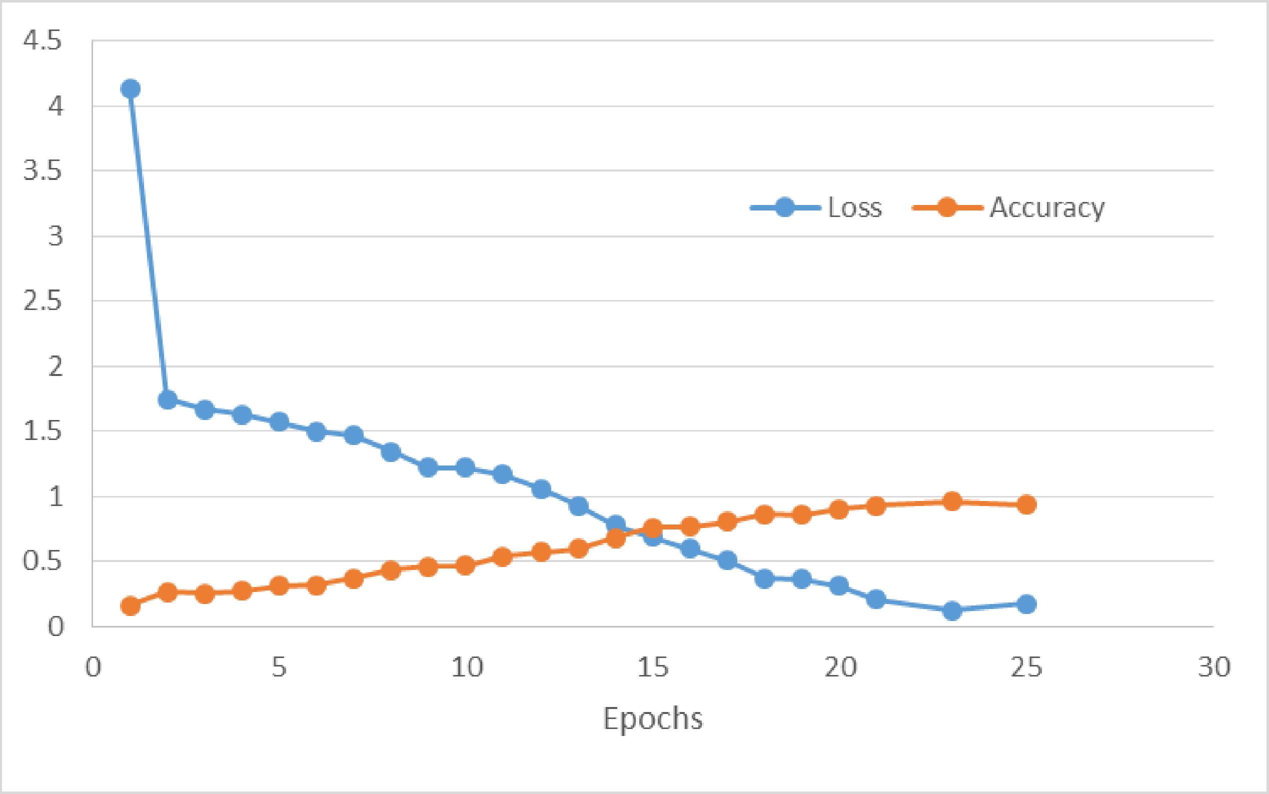
- Decision Tree



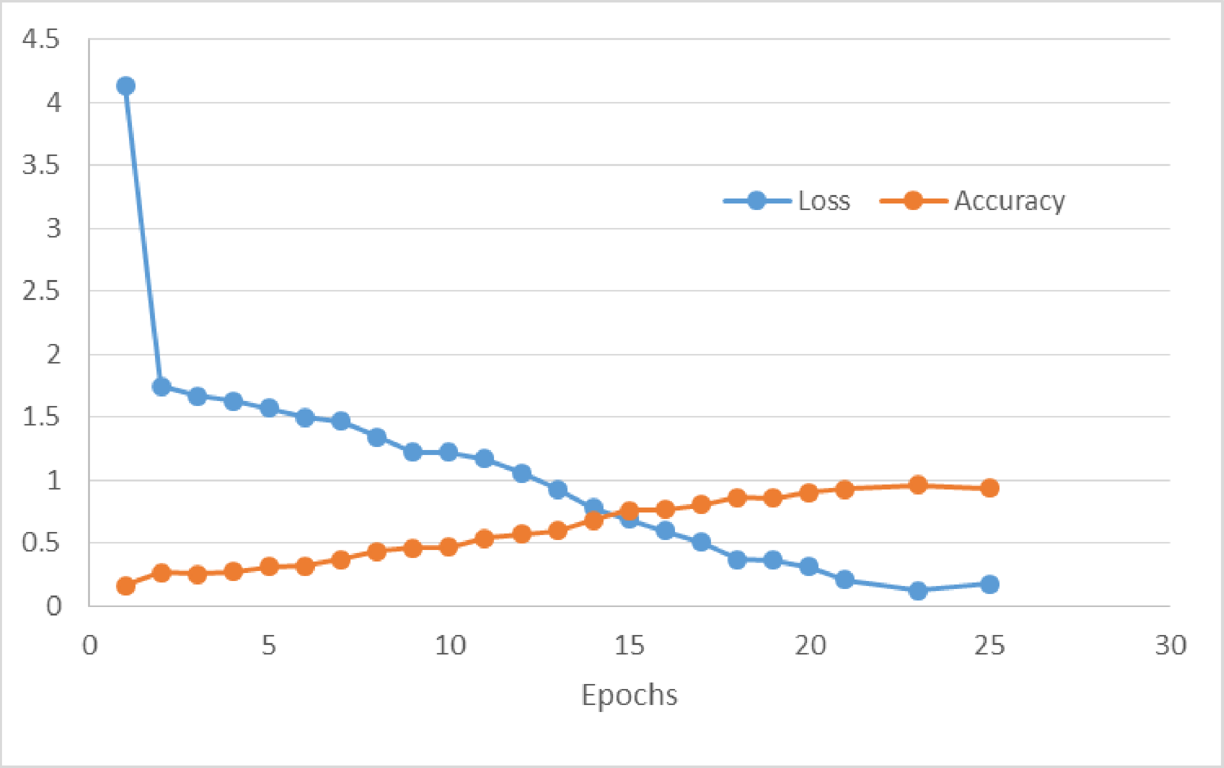
- Random Tree Forest



- MLP



- CNN



11.2 Classification Report

1. Augmented Data (Increased)

```
Classification Report decision tree:
              precision    recall  f1-score   support

           0       0.99      0.98      0.99      1730
           1       0.98      0.99      0.98      1750
           2       0.93      1.00      0.96      1689
           3       1.00      1.00      1.00      1907
           4       0.98      0.76      0.86      1683
           5       0.92      1.00      0.96      1924
           6       0.98      1.00      0.99      1697

    accuracy                           0.96     12380
   macro avg       0.97      0.96      0.96     12380
weighted avg       0.97      0.96      0.96     12380
```

```
Classification Report random tree:
              precision    recall  f1-score   support

           0       0.81      1.00      0.90      1730
           1       1.00      1.00      1.00      1750
           2       0.99      1.00      1.00      1689
           3       1.00      1.00      1.00      1907
           4       1.00      0.74      0.85      1683
           5       0.99      1.00      0.99      1924
           6       1.00      1.00      1.00      1697

    accuracy                           0.96     12380
   macro avg       0.97      0.96      0.96     12380
weighted avg       0.97      0.96      0.96     12380
```

2. Augmented Data (Decreased)

```
 Classification Report decision tree:
              precision    recall  f1-score   support

           0       0.50      0.60      0.55        25
           1       0.33      0.32      0.33        28
           2       0.58      0.52      0.55        27
           3       0.37      0.37      0.37        27
           4       0.39      0.29      0.33        24
           5       0.41      0.52      0.46        21
           6       0.45      0.43      0.44        23

    accuracy                           0.43       175
   macro avg       0.43      0.44      0.43       175
weighted avg       0.43      0.43      0.43       175
```

```
Classification Report random tree:
              precision    recall  f1-score   support

           0       0.17      1.00      0.30        25
           1       1.00      0.07      0.13        28
           2       0.75      0.22      0.34        27
           3       1.00      0.04      0.07        27
           4       0.83      0.21      0.33        24
           5       0.40      0.19      0.26        21
           6       1.00      0.17      0.30        23

    accuracy                           0.27       175
   macro avg       0.74      0.27      0.25       175
weighted avg       0.75      0.27      0.24       175
```

```
Classification Report knn:
              precision    recall  f1-score   support

           0       0.14      1.00      0.25        25
           1       0.00      0.00      0.00        28
           2       1.00      0.04      0.07        27
           3       0.00      0.00      0.00        27
           4       0.00      0.00      0.00        24
           5       0.00      0.00      0.00        21
           6       0.00      0.00      0.00        23

    accuracy                           0.15       175
   macro avg       0.16      0.15      0.05       175
weighted avg       0.17      0.15      0.05       175
```

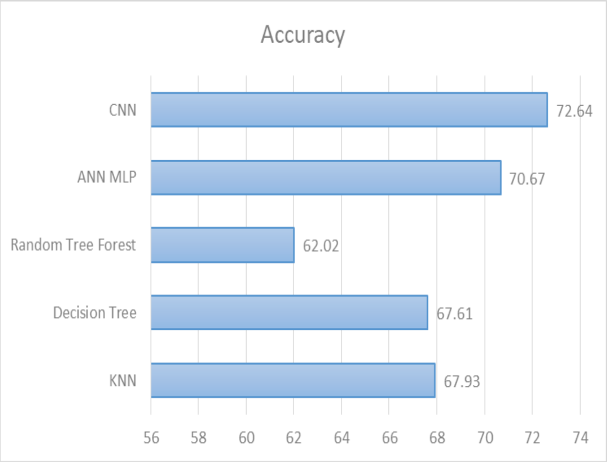
3. Original Data

```
Classification Report decision treee:
              precision    recall  f1-score   support

           0       0.17      0.18      0.17        68
           1       0.24      0.24      0.24       123
           2       0.31      0.34      0.32       277
           3       0.04      0.03      0.04        30
           4       0.82      0.80      0.81      1693
           5       0.27      0.28      0.27       274
           6       0.07      0.05      0.06        39

    accuracy                           0.63      2504
   macro avg       0.27      0.28      0.27      2504
weighted avg       0.63      0.63      0.63      2504
```

```
Classification Report random tree:
               precision    recall  f1-score   support

           0       0.09      0.93      0.16        68
           1       0.88      0.06      0.11       123
           2       0.62      0.09      0.16       277
           3       0.00      0.00      0.00        30
           4       0.86      0.86      0.86      1693
           5       0.73      0.03      0.06       274
           6       0.00      0.00      0.00        39

    accuracy                           0.62      2504
   macro avg       0.45      0.28      0.19      2504
weighted avg       0.77      0.62      0.61      2504

Classification Report Knn:
               precision    recall  f1-score   support

           0       0.09      0.69      0.16        68
           1       0.00      0.00      0.00       123
           2       0.00      0.00      0.00       277
           3       0.00      0.00      0.00        30
           4       0.78      0.93      0.85      1693
           5       0.00      0.00      0.00       274
           6       0.00      0.00      0.00        39

    accuracy                           0.64      2504
   macro avg       0.13      0.23      0.14      2504
weighted avg       0.53      0.64      0.58      2504
```

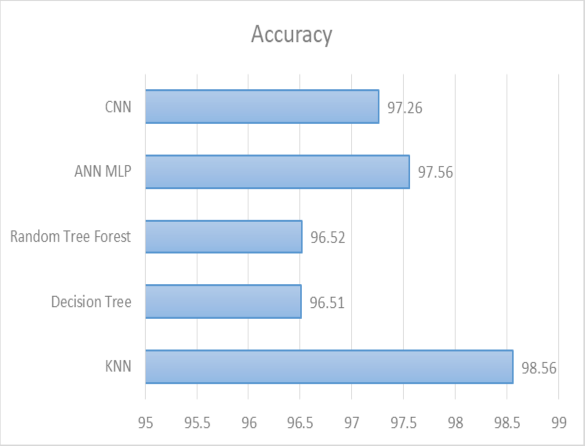
11.3 Model Evaluation

1. Original Data



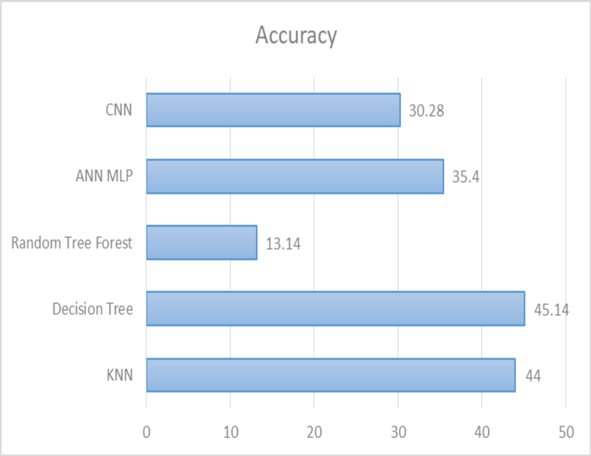
Algorithms	Accuracy
KNN	67.93
Decision Tree	67.61
Random Tree Forest	62.02
ANN MLP	70.67
CNN	72.64

2. Augmented Data (Increased)



Algorithms	Accuracy
KNN	98.56
Decision Tree	96.51
Random Tree Forest	96.52
ANN MLP	97.56
CNN	97.26

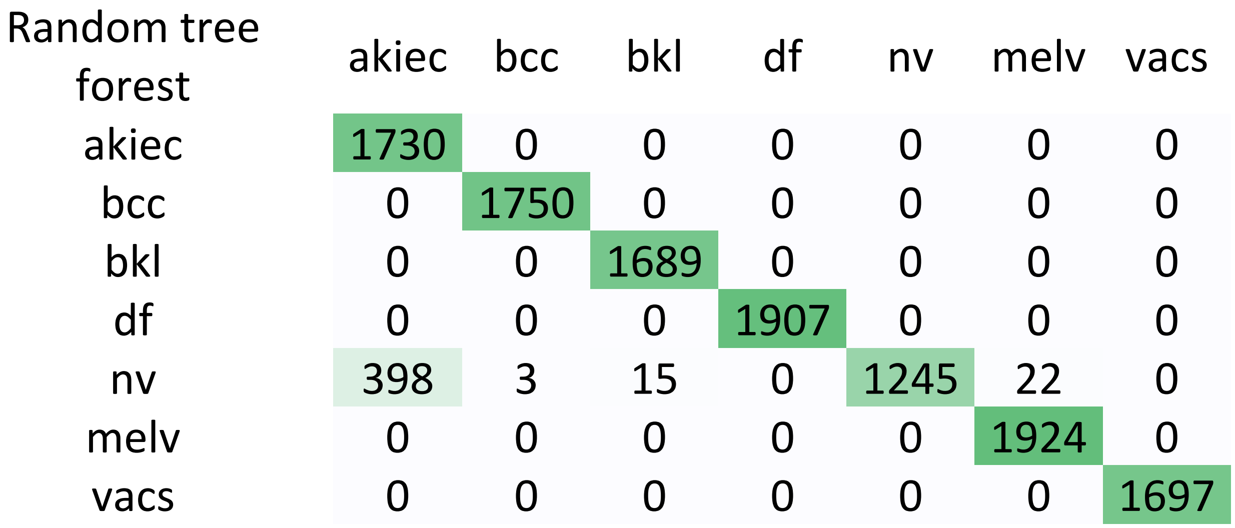
3. Augmented Data (Decreased)



Algorithms	Accuracy
KNN	44
Decision Tree	45.14
Random Tree Forest	13.14
ANN MLP	35.40
CNN	30.28

11.4 Confusion Matrix

1. Augmented Data (Increased)



Random tree forest	akiec	bcc	bkl	df	nv	melv	vacs
akiec	1730	0	0	0	0	0	0
bcc	0	1750	0	0	0	0	0
bkl	0	0	1689	0	0	0	0
df	0	0	0	1907	0	0	0
nv	398	3	15	0	1245	22	0
melv	0	0	0	0	0	1924	0
vacs	0	0	0	0	0	0	1697

Decision Tree	akiec	bcc	bkl	df	nv	melv	vacs
akiec	1720	0	7	0	21	0	0
bcc	0	1736	0	0	11	0	3
bkl	0	0	1689	0	0	0	0
df	0	0	0	1907	0	0	0
nv	22	41	126	7	1284	176	27
melv	0	0	2	0	0	1922	0
vacs	0	0	0	0	0	0	1697

KNN	akiec	bcc	bkl	df	nv	melv	vacs
akiec	1242	114	35	254	6	0	79
bcc	834	485	48	290	0	5	88
bkl	871	69	357	259	30	53	50
df	511	27	0	1369	0	0	0
nv	610	26	35	164	695	58	95
melv	1226	6	143	112	46	297	94
vacs	319	12	0	70	39	0	1257

2. Original Data

Decision Tree	akiec	bcc	bkl	df	nv	melv	vacs
akiec	12	15	14	3	18	5	1
bcc	15	30	25	6	34	9	4
bkl	12	21	94	6	102	41	1
df	3	5	6	1	11	3	1
nv	18	37	111	10	1357	149	11
melv	8	14	49	0	118	77	8
vacs	3	4	7	0	20	3	2

Random Forest	akiec	bcc	bkl	df	nv	melv	vacs
akiec	63	1	0	0	4	0	0
bcc	99	7	2	0	15	0	0
bkl	164	0	25	0	88	0	0
df	23	0	0	0	7	0	0
nv	221	0	10	0	1459	3	0
melv	149	0	3	0	114	8	0
vacs	21	0	0	0	18	0	0

KNN	akiec	bcc	bkl	df	nv	melv	vacs
akiec	47	0	0	0	21	0	0
bcc	80	0	0	0	43	0	0
bkl	136	0	0	0	141	0	0
df	14	0	0	0	16	0	0
nv	126	0	0	0	1567	0	0
melv	90	0	0	0	184	0	0
vacs	11	0	0	0	28	0	0

3. Augmented Data (Decreased)

Desion tree	akiec	bcc	bkl	df	nv	melv	vacs
akiec	15	3	0	1	2	2	2
bcc	4	9	1	7	2	3	2
bkl	0	1	14	3	1	5	3
df	6	4	2	10	2	1	2
nv	3	6	1	2	7	2	3
melv	1	2	3	2	2	11	0
vacs	1	2	3	2	2	3	10

RT	akiec	bcc	bkl	df	nv	melv	vacs
akiec	25	0	0	0	0	0	0
bcc	25	2	0	0	1	0	0
bkl	15	0	6	0	0	6	0
df	26	0	0	1	0	0	0
nv	19	0	0	0	5	0	0
melv	15	0	2	0	0	4	0
vacs	19	0	0	0	0	0	4

KNN	akiec	bcc	bkl	df	nv	melv	vacs
akiec	25	0	0	0	0	0	0
bcc	28	0	0	0	0	0	0
bkl	25	0	1	0	0	1	0
df	27	0	0	0	0	0	0
nv	24	0	0	0	0	0	0
melv	21	0	0	0	0	0	0
vacs	23	0	0	0	0	0	0

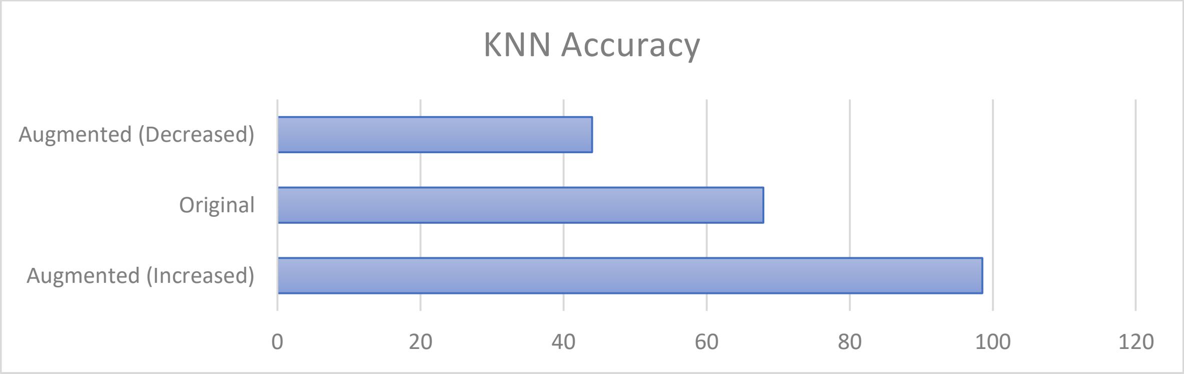
11.5 Comparison

Our first stage was the comparison of datasets accuracy in which we showed which dataset gave high accuracy then later on we started comparing according to our founding scope of the project which is MLP vs Neural Network.

Performance of KNN is as follows

KNN	
Data Set	Accuracy
Augmented (Increased)	98.56
Original	67.93
Augmented (Decreased)	44
Average Performance	70.16

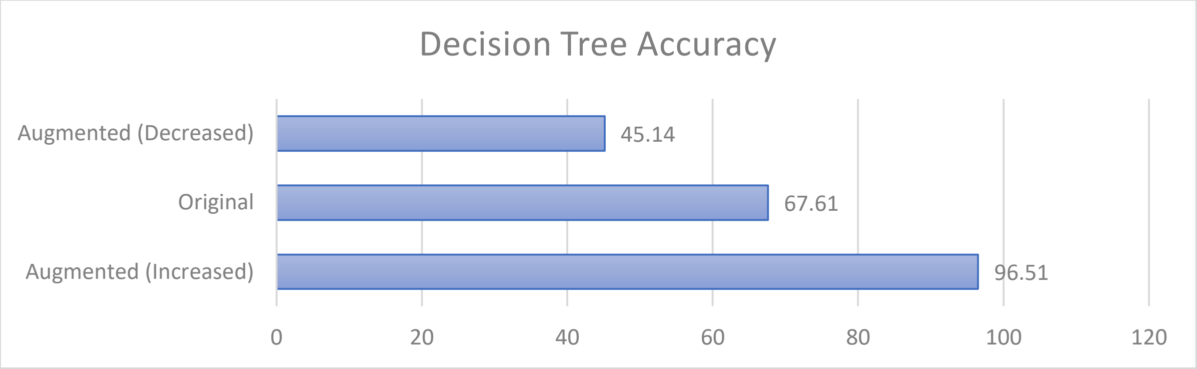
So as we can see in above the KNN has outperformed in the Increased dataset version has when we took a general average of KNN it has showed us that the KNN is much better and consistent throughout the data evaluation.



Performance of Decision Tree is as follows

Decision Tree	
Data Set	Accuracy
Augmented (Increased)	96.51
Original	67.61
Augmented (Decreased)	45.14
Average Performance	69.75

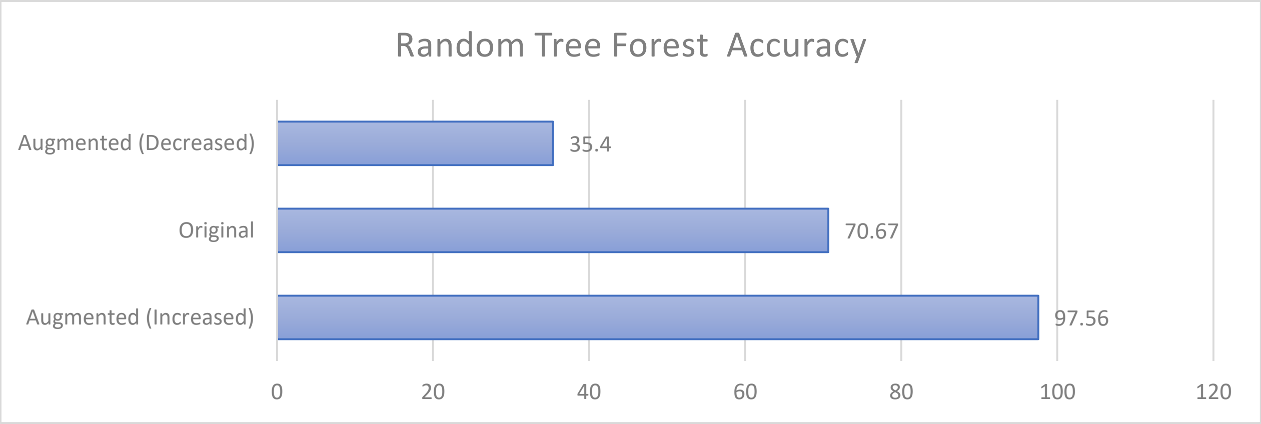
So as we can see in above the Decision Tree has averagely performed as when we took a general average of it has showed us that the KNN is much better and consistent throughout the data evaluation. Decision tree was the second most successful algorithm after KNN.



<u>Performance of Random Forest Tree is as follows</u>

Random Tree Forest	
Data Set	Accuracy
Augmented (Increased)	97.56
Original	70.67
Augmented (Decreased)	35.40
Average Performance	67.87

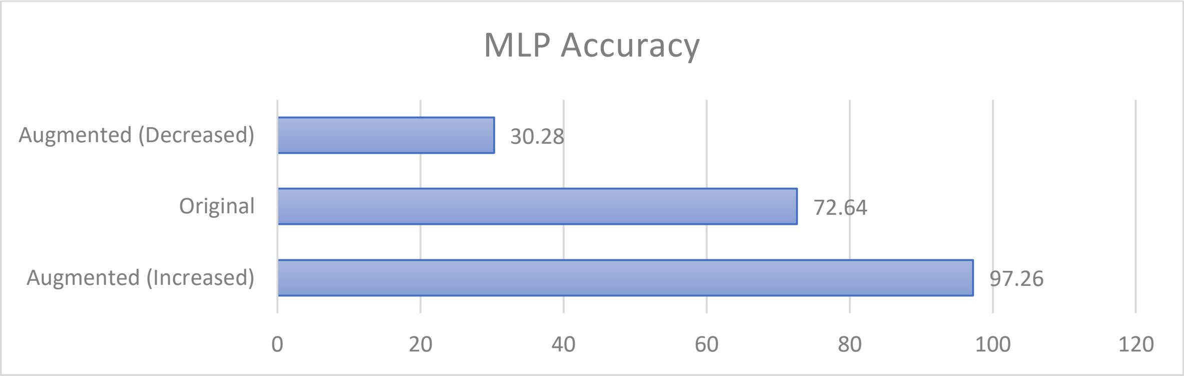
*So the effect of the fluctuating result can be seen here that shows us that the algorithm has performed under the par against its predecessors but on the contrary the random forest has overall got the 2nd number in highest accuracy average***.**



Performance of MLP is as follows

MLP	
Data Set	Accuracy
Augmented (Increased)	97.26
Original	72.64
Augmented (Decreased)	30.28
Average Performance	66.72

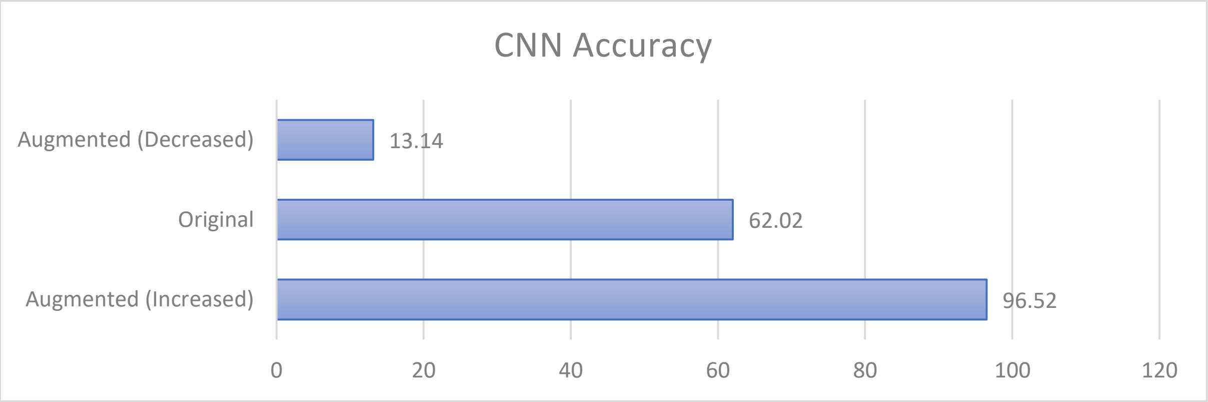
So MLP is the first algorithm of neural coming up the problem here with it was that it was not working well with the less optimized dataset but its considered to be a very intelligent algorithm which performed very well in tough situations as un-adjusted dataset where data wasn't equal.



Performance of CNN is as follows

CNN	
Data Set	Accuracy
Augmented (Increased)	96.52
Original	62.02
Augmented (Decreased)	13.14
Average Performance	57.22

CNN and MLP if you look from different perspective then these two were performing well on the unadjusted and increased version but the decreased data brought their average accuracy by a lot of margin hence causing a downfall in the ranking of the ML and NN.

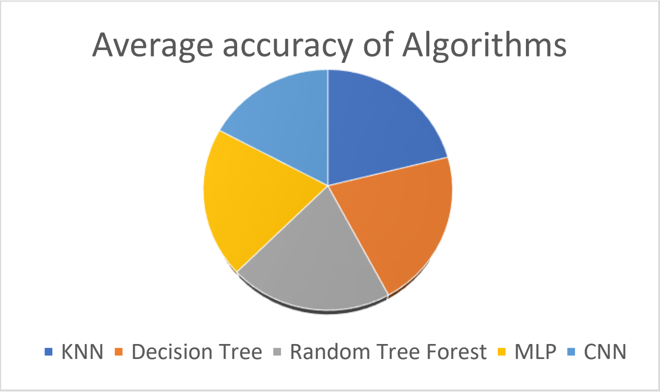


Concluding Comparison

Now if we take an average of all the 5 Algorithms things become clear which algorithm works more effectively that which algorithm had worked and competed with its counterparts. Which Algorithm had been more vigilant in obtaining the highest accuracy hence we can easily obtain this data by comparing the average accuracy of all the 5 algorithms side by side and analyzing it deeply. This analysis of algorithms will help us understanding that which algorithm was the most useful for cancer detection and for which particular algorithms we need to provide detailed adjusted data

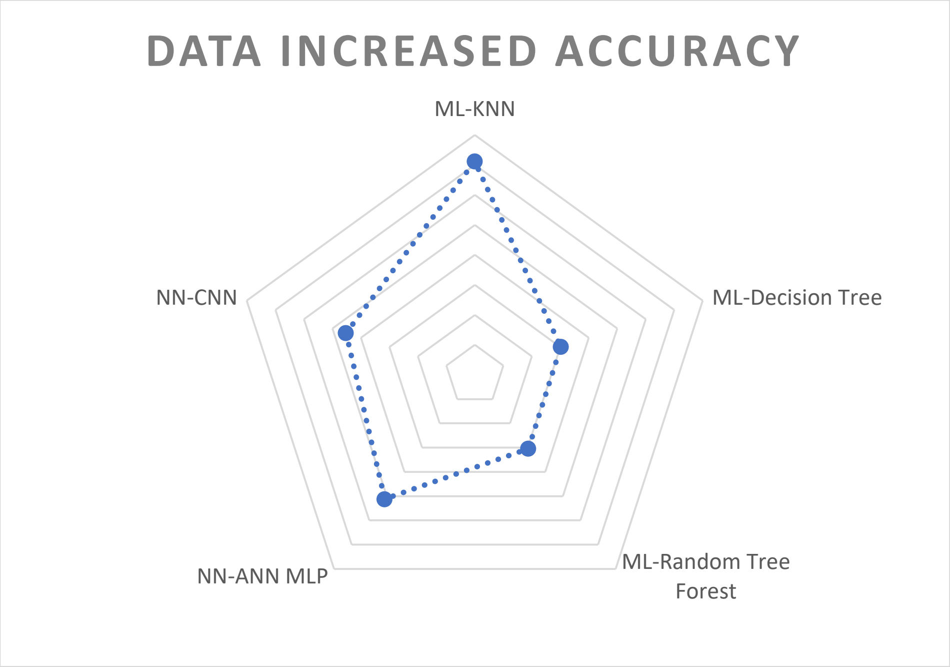
Below is the average of all algorithm's accuracy in graphical form:-

Algorithms	Average Accuracy
KNN	70.16
Decision Tree	69.75
Random Tree Forest	67.87
MLP	66.72
CNN	57.22

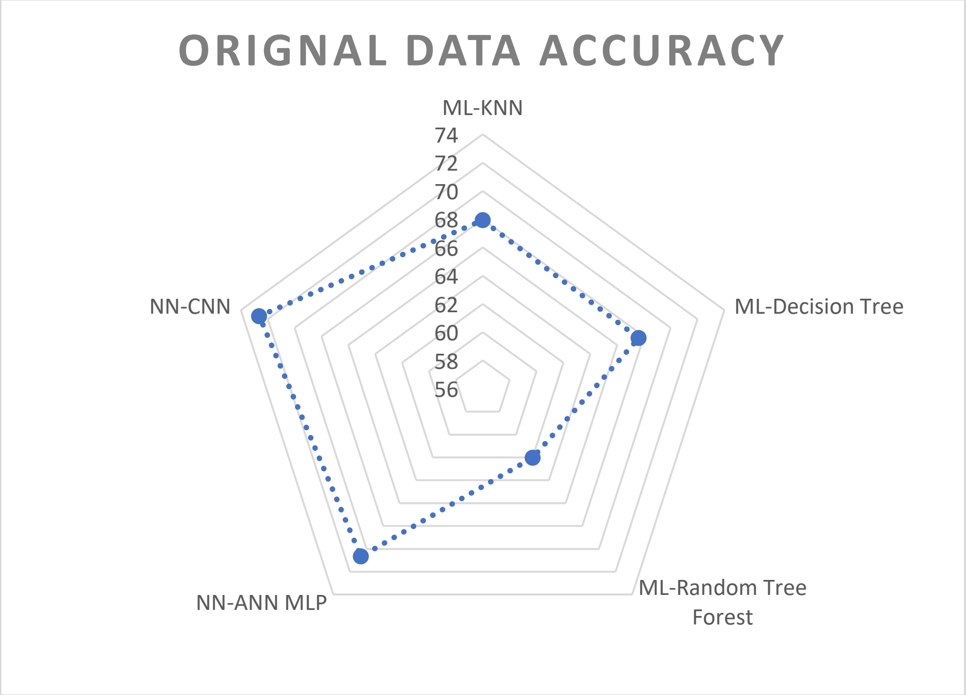


12. Observation

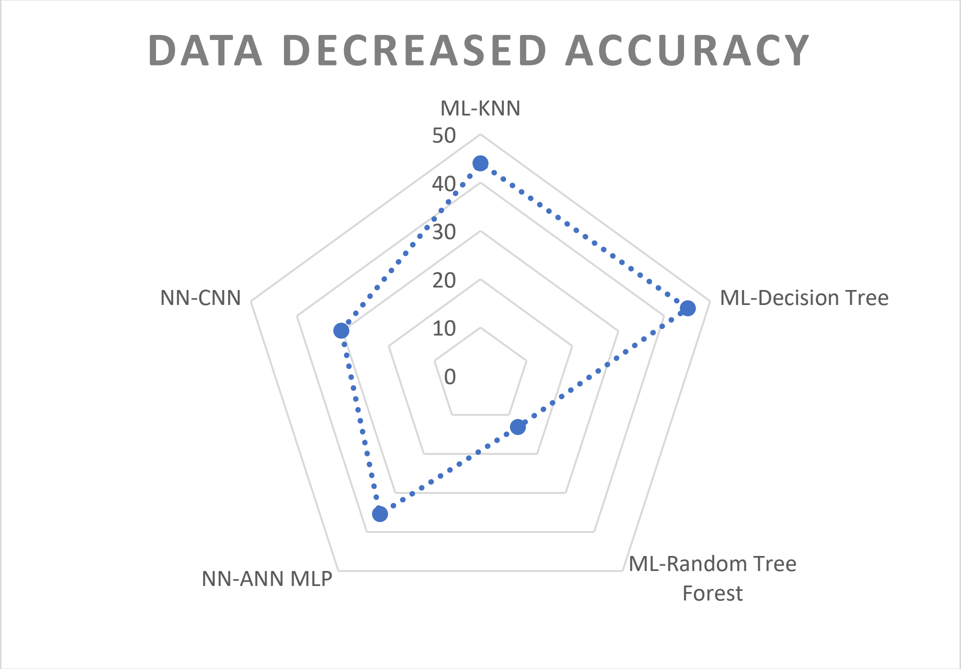
So our outcome of this project is that the MLP algorithms had worked much better with the given dataset. Neural or deep learning algorithms had a terrible accuracy, the ideal reason is because that the Neural and Deep learning Algorithms are made to be implemented efficiently on large dataset so that it can perform much better but the given above dataset was not that much big, but when we increased the dataset as you can clearly see that in 50K the dataset clearly gave High accuracy but in 10k and decreased dataset which was 700 it gave terrible accuracy.



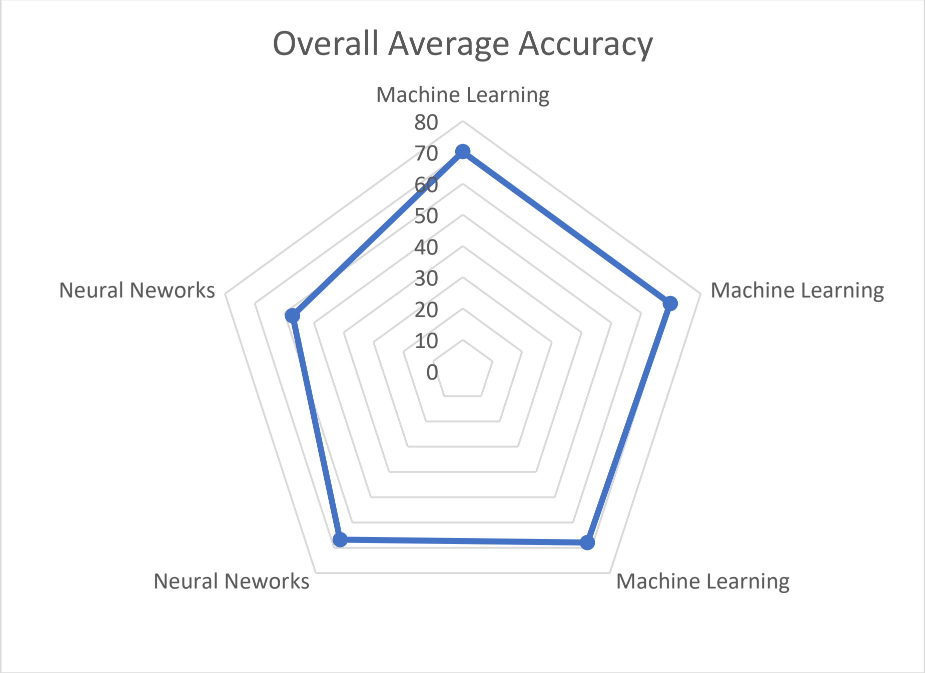
So if we conclude the computing data where we increased the data there as the graph itself speaks we can see that end-point of ML algorithms are more towards the edge. This generally means that they are the optimal algorithm for it but the difference in this was not that much it was generally in 2 or 1 MSD so this section alone cannot give or decide that which algorithm was best for the detection.



As I have said earlier that Neural Network algorithms are much better and highly advanced in testing where as its competitor were not able to handle the mismatched data hence the performed very badly in this section. So this graph supports our takeaway that Neural Network is much better in identifying and work in datasets with ambiguity, anomaly or imbalance. As the neural network are generally more than just taking quantitate values hence they showed their capability here.

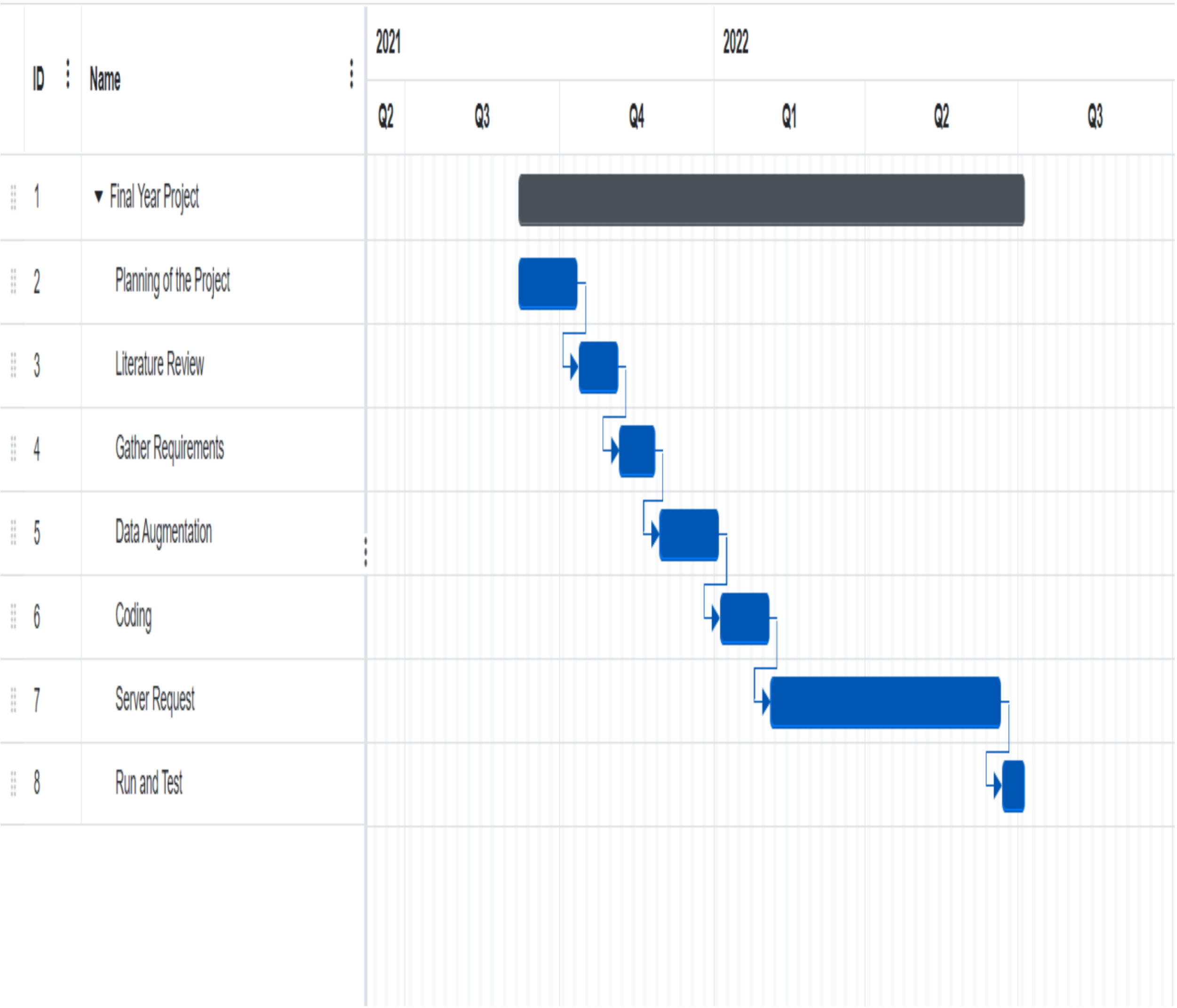


As we can clearly see how the ML end points are more toward their boundaries as compared to NN this clearly show that here ML is outperforming the Neural Network. This doesn't support the Idea that neural networks can be deemed as weak on this datasets but they are made to play on data which is unadjusted and etc. But ML is considered more powerful when provided level playing field.



The above graphical representation has shown that the ML algorithms are much better for as compared to NN algorithms as they are more consistent where as compared to NN their accuracy is fluctuating more as compared to others. But this doesn’t mean that NN is not suitable here but ML is far more consistent with result in this as Neural Network is suitable for data which is not that corrected however the question is that Neural Network will work provided any condition but here on this dataset ML is suitable

12. Gantt Chart



ID	Name
1	▾ Final Year Project
2	Planning of the Project
3	Literature Review
4	Gather Requirements
5	Data Augmentation
6	Coding
7	Server Request
8	Run and Test

13. Future Advancement

We would be developing a front-end section of the system which would be helping the doctors run the system easily and other people who are not that familiar with back-end development.

Also after sometime we would be adding more real pictures to make the data train more accurately and increasing the accuracy.

In future work if we want to get our project of skin cancer detection as reality then backend should be made where detection is done by the neural network IF we assume data cannot be provided consistently and equally. If we think that data cannot be provided equally so neural is ideal BUT if we know Data from which is going to compare our picture that data is even and optimized then ML should be used in any project that is dealing with skin cancer detection.

Conclusion

In this project, we worked on a variety of machine learning and deep learning algorithms and conducted a two-phase comparative analysis. The first phase includes seven different classes of patient i.e. Melanoma , Melanocytic nevi , Benign keratosis-like lesions , Basal cell carcinoma , Actinic keratosis, Vascular lesions, Dermatofibroma . We will examine the optimal method for the testing phase outcome by executing the selected algorithms on the specified dataset. This project will be implemented using Python and its libraries as a tool. The sort of skin cancer we found differs as a result of our findings.

The 2nd phase had included that we analyze the data first reducing to just 800 then running the algorithm and seeing how it actually performed. Then Running the algorithms on the dataset which is original which had anomalies had discussed above still considering them and running algorithms. So the takeaway was here neural network was much better as it is detailed algorithm having capabilities to identify any problems or redundancies which it did hence it was the most high performing algorithm among all.t Then we increased the dataset multiplying each disease category by such number which brought each categories data as equal. Here ML was better but with little difference compared with neural networks.

So the key take away we can say is that neural network was much better as compared to others only in condition where data was not optimized and cleansed hence if working with uncleansed data only algorithm which can give us optimal and healthy result is neural network.

ML was not that much good with the uncleansed data or uneven data therefore we can say in playing even field the ML algorithms can do fairly well as compared without any fluctuations.

Reference

Machine Learning

[1.] Mitchell, Tom (1997). Machine Learning. New York: McGraw Hill. ISBN 0-07-042807-7. OCLC 36417892.

[2]. The definition "without being explicitly programmed" is often attributed to Arthur Samuel, who coined the term "machine learning" in 1959, but the phrase is not found verbatim in this publication, and may be a paraphrase that appeared later. Confer "Paraphrasing Arthur Samuel (1959), the question is: How can computers learn to solve problems without being explicitly programmed?" in Koza, John R.; Bennett, Forrest H.; Andre, David; Keane, Martin A. (1996). Automated Design of Both the Topology and Sizing of Analog Electrical Circuits Using Genetic Programming. Artificial Intelligence in Design '96. Springer, Dordrecht. pp. 151–170. Doi: 10.1007/978-94-009-0279-4_9.

[3] Hu, J.; Nia, H.; Carrasco, J.; Lennox, B.; Arvin, F., "Coronoid-Based Multi-Robot Autonomous Exploration in Unknown Environments via Deep Reinforcement Learning" IEEE Transactions on Vehicular Technology, 2020.

*[4] Jump up to: ***a b c d*** Bishop, C. M. (2006), Pattern Recognition and Machine Learning, Springer, ISBN 978-0-387-31073-2*

[5] Machine learning and pattern recognition "can be viewed as two facets of the same field."[4]:vii

*[6] Friedman, Jerome H. (1998). "Data Mining and Statistics: What's the connection?" Computing Science and Statistics. **29** (1): 3–9.*

[7]. "What is Machine Learning?" www.ibm.com. Retrieved 2021-08-15.

[8]. Zhou, Victor (2019-12-20). "Machine Learning for Beginners: An Introduction to Neural Networks". Medium. Retrieved 2021-08-15.

Supervised Leaning

[9]. Russell, Stuart J.; Norvig, Peter (2010). Artificial Intelligence: A Modern Approach (Third Ed.). Prentice Hall. ISBN 9780136042594.

[10]. Mohri, Mehryar; Rostamizadeh, Afshin; Talwalkar, Ameet (2012). Foundations of Machine Learning. The MIT Press. ISBN 9780262018258.

Unsupervised leaning

[11]. Jordan, Michael I.; Bishop, Christopher M. (2004). "Neural Networks". In Allen B. Tucker (Ed.). Computer Science Handbook, Second Edition (Section VII: Intelligent Systems). Boca Raton, Florida: Chapman & Hall/CRC Press LLC. ISBN 978-1-58488-360-9.

Reinforcement learning

[12]. Van Otterlo, M.; Wiering, M. (2012). Reinforcement learning and markov decision processes. Reinforcement Learning. Adaptation, Learning, and Optimization. ***12****. pp. 3–42. Doi: 10.1007/978-3-642-27645-3_1. ISBN 978-3-642-27644-6.*

Deep learning

[13]. Bengio, Y.; Courville, A.; Vincent, P. (2013). "Representation Learning: A Review and New Perspectives". IEEE Transactions on Pattern Analysis and Machine Intelligence. ***35*** *(8): 1798–1828. ArXiv: 1206.5538. doi:10.1109/tpami.2013.50. PMID 23787338. S2CID 393948.*

[14]. Jump up to: [a] [b] [c] [d] [e] [f] [g] [h] Schmidhuber, J. (2015). "Deep Learning in Neural Networks: An Overview". Neural Networks. ***61****: 85–117. ArXiv: 1404.7828. doi:10.1016/j.neunet.2014.09.003. PMID 25462637. S2CID 11715509.*

[15]. *Bengio, Yoshua; LeCun, Yann; Hinton, Geoffrey (2015). "Deep Learning". Nature.* ***521*** *(7553): 436–444. Bibcode: 2015Natur.521...436L. Doi: 10.1038/nature14539. PMID 26017442. S2CID 3074096.*

[16]. *Hu, J.; Niu, H.; Carrasco, J.; Lennox, B.; Arvin, F. (2020). "Voronoi-Based Multi-Robot Autonomous Exploration in Unknown Environments via Deep Reinforcement Learning". IEEE Transactions on Vehicular Technology.* ***69*** *(12): 14413–14423. doi:10.1109/TVT.2020.3034800. S2CID 228989788. Archived from the original on 2020-11-16. Retrieved 2021-05-04.*

[17]. Jump up to: [a] [b] Ciresan, D.; Meier, U.; Schmidhuber, J. (2012). "Multi-column deep neural networks for image classification". 2012 IEEE Conference on Computer Vision and Pattern Recognition. pp. 3642–3649. ArXiv: 1202.2745. doi:10.1109/cvpr.2012.6248110. ISBN 978-1-4673-1228-8. S2CID 2161592.

[18]. Jump up to: [a] [b] Krizhevsky, Alex; Sutskever, Ilya; Hinton, Geoffry (2012). "ImageNet Classification with Deep Convolutional Neural Networks" (PDF). NIPS 2012: Neural Information Processing Systems, Lake Tahoe, Nevada. Archived (PDF) from the original on 2017-01-10. Retrieved 2017-05-24.

[19]. *"Google's AlphaGo AI wins three-match series against the world's best Go player". TechCrunch. 25 May 2017. Archived from the original on 17 June 2018. Retrieved 17 June 2018.*

[20]. *Marblestone, Adam H.; Wayne, Greg; Kording, Konrad P. (2016). "Toward an Integration of Deep Learning and Neuroscience". Frontiers in Computational Neuroscience.* ***10****: 94. ArXiv: 1606.03813. Bibcode: 2016arXiv160603813M. doi:10.3389/fncom.2016.00094. PMC 5021692. PMID 27683554. S2CID 1994856.*

__[21].__ Olshausen, B. A. (1996). "Emergence of simple-cell receptive field properties by learning a sparse code for natural images". Nature. __381__ (6583): 607–609. Bibcode: 1996Natur.381...607O. Doi: 10.1038/381607a0. PMID 8637596. S2CID 4358477.

__[22].__ Bengio, Yoshua; Lee, Dong-Hyun; Bornschein, Jorg; Mesnard, Thomas; Lin, Zhouhan (13 February 2015). "Towards Biologically Plausible Deep Learning". ArXiv: 1502.04156 [cs.LG].

ANN

[24]. Bengio, Yoshua (2009). "Learning Deep Architectures for AI" (PDF). Foundations and Trends in Machine Learning. __2__ (1): 1–127. CiteSeerX 10.1.1.701.9550. Doi: 10.1561/2200000006. Archived from the original (PDF) on 4 March 2016. Retrieved 3 September 2015.

[24]. A Guide to Deep Learning and Neural Networks, archived from the original on 2020-11-02, retrieved 2020-11-16

RNN

[25]. Gers, Felix A.; Schmidhuber, Jürgen (2001). "LSTM Recurrent Networks Learn Simple Context Free and Context Sensitive Languages". IEEE Transactions on Neural Networks. __12__ (6): 1333–1340. Doi: 10.1109/72.963769. PMID 18249962. Archived from the original on 2020-01-26. Retrieved 2020-02-25.

[26]. Jump up to: [a] [b] [c] Sutskever, L.; Vinyl's, O.; Le, Q. (2014). "Sequence to Sequence Learning with Neural Networks" (PDF). Proc. NIPS. ArXiv: 1409.3215. Bibcode: 2014arXiv1409.3215S. Archived (PDF) from the original on 2021-05-09. Retrieved 2017-06-13.

[27]. Jump up to: [a] [b] Jozefowicz, Rafal; Vinyals, Oriol; Schuster, Mike; Shazeer, Noam; Wu, Yonghui (2016). "Exploring the Limits of Language Modeling". ArXiv: 1602.02410 [cs.CL].

[28]. Jump up to: [a] [b] Gillick, Dan; Brunk, Cliff; Vinyals, Oriol; Subramanya, Amarnag (2015). "Multilingual Language Processing from Bytes". ArXiv: 1512.00103 [cs.CL].

[29]. Mikolov, T.; et al. (2010). "Recurrent neural network based language model" (PDF). Interspeech: 1045–1048. Doi: 10.21437/Interspeech.2010-343. Archived (PDF) from the original on 2017-05-16. Retrieved 2017-06-13.

CNN

[30]. LeCun, Y.; et al. (1998). "Gradient-based learning applied to document recognition". Proceedings of the IEEE. __86__ (11): 2278–2324. Doi: 10.1109/5.726791.

[31]. Sainath, Tara N.; Mohamed, Abdel-Rahman; Kingsbury, Brian; Ramabhadran, Bhuvana (2013). "Deep convolutional neural networks for LVCSR". 2013 IEEE International Conference on Acoustics,

Speech and Signal Processing. pp. 8614–8618. Doi: 10.1109/icassp.2013.6639347. ISBN 978-1-4799-0356-6. S2CID 13816461.

Data Set

[32]. Rembielak A, Ajithkumar T (2019) Non-melanoma skin cancer–an underestimated global health threat. Clin Oncol 31(11):735–737

[33]. Tschandl, C. Rosendahl, and H. Kittler, "The HAM10000 Dataset, a Large Collection of Multi-Source Dermatoscopic Images of Common Pigmented Skin Lesions," Scientific data, vol. 5, p. 180161, 2018.