NETAJI SUBHAS INSTITUTE OF TECHNOLOGY, BIHTA, PATNA

INTERNSHIP REPORT

ON

PREDICTION OF FOLIAR DISEASES IN APPLE LEAVES

SUBMITTED IN THE PARTIAL FULFILLMENT OF THE REQUIREMENT

OF

BACHELOR OF TECHNOLOGY (COMPUTER SCIENCE AND ENGINEERING)

BY

ARYABHATTA KNOWLEDGE UNIVERSITY, PATNA, BIHAR

SUBMITTED BY:

Swadha Kumari (181029)

Karan Kumar (181040)

Priti Kumari (181038)

Vth sem

SESSION 2018-22

An ISO 9001:2008 Certified Institution

Approved by AICTE, New Delhi

Affiliated to Aryabhatta Knowledge University, Patna, Bihar

NETAJI SUBHAS INSTITUTE OF TECHNOLOGY, BIHTA, PATNA

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SESSION- 2018-22

ACKNOWLEDGEMENT

The internship opportunity, we had with IIIT Dharwad, was a great chance for learning and professional development. Therefore, we consider ourselves lucky and would like to express our deepest gratitude to Dr. Sunil Saumya, Assistant Professor, IIIT Dharwad, who inspite of having a busy schedule, heartily guided and encouraged us throughout the internship.

We would also like to convey our deepest gratitude to Mr. Gopal Krishna, Assistant Professor, NSIT Patna, for taking part in useful decisions and giving necessary advice and guidance and also for arranging all facilities to make the training experience easier.

Last but not the least, we would like to thank our family and friends, for all that they meant to us during the crucial times of the completion of our internship.

Swadha Kumari (181029)

Karan Kumar (181040)

Priti Kumari (181038)

ABOUT THE ORGANIZATION

The Indian Institute of Information Technology Dharwad is one of the 20 IIITs proposed under non-profit, Public-Private Partnership model set up by the Ministry of Human Resource Development (MHRD), Government of India. It is an academic and research institute funded by the Government of India, Government of Karnataka and industrial partner Keonics. The institute has been declared as an Institute of National Importance under the Indian Institutes of Information Technology Public-Private Partnership Act of Parliament (No. 23 of 2017).

The Institute commenced its academic session from August 2015. The Institute is currently operating out of its transit campus housed in the IT Park in Hubballi with well-equipped classrooms, laboratories and a library.IIIT Dharwad is keen to develop its permanent campus in a plot of 61 acres allotted by the Government of Karnataka at Tadasinakoppa village between Hubballi and Dharwad cities. As a not-for-profit institute of higher learning in the area of Information Technology (IT), the primary objective of IIIT Dharwad is to address the skill gap in high-end information technology and thereby enable India to retain its global leadership role in IT and allied areas. To achieve this objective, IIIT Dharwad currently offers B.Tech. Degree programmes in Computer Science and Electronics Communication.

One of the key strengths of this young Institute is its highly qualified and committed faculty with expertise in a wide range of areas of information technology and with PhDs and research as well as industry experience from reputed institutes in India and abroad. As a young institute, IIIT Dharwad has a unique opportunity to make a difference not only to Indian IT industry and the academic research community, but also the people of the entire region of North Karnataka. The twin cities of Hubballi and Dharwad are already a recognized centre for educational institutes of repute including engineering, medical, law and other colleges, universities and an Indian Institute of Technology. With this environment and rapidly improving connectivity to the IT capital Bengaluru, IIIT Dharwad is strategically positioned to develop into one of the best institutes of national importance.

Contents

Abstract **1**

1 INTRODUCTION **2**

2 RELATED WORK **3**

3 PRELIMINARIES **5**

3.1 MACHINE LEARNING 5

3.1.1 NEED 6

3.1.2 WORKING 6

3.1.3 CHARACTERISTICS 7

3.1.4 APPLICATIONS 8

3.1.5 CLASSIFICATION 10

3.1.6 SUPERVISED MACHINE LEARNING 11

3.1.6.1 Regression 11

3.1.6.2 Classification 12

3.1.7 UNSUPERVISED MACHINE LEARNING 15

3.1.7.1 Clustering 15

3.1.7.2 Association 17

3.1.8 REINFORCEMENT MACHINE LEARNING 17

3.1.8.1 Positive Reinforcement 17

3.1.8.2 Negative Reinforcement 18

3.2 DEEP LEARNING 18

3.2.1 APPLICATIONS 18

3.2.2 NEURAL NETWORK 19

3.2.2.1 Application 19

3.2.3 CONVOLUTIONAL NEURAL NETWORK 20

3.2.3.1 why convnets over feed-forward neural nets 20

3.2.3.2 input image 20

3.2.4 RECURRENT NEURAL NETWORK 21

4 PREPROCESSING TECHNIQUES REQUIRED **22**
4.1 READING IMAGES 23
4.2 RESIZING IMAGE 23
4.3 DENOISING IMAGE 24
4.4 SEGMENTATION AND MORPHOLOGY 26
4.5 EDGE DETECTION 27

5 PROPOSED METHODOLOGY **29**
5.1 DATASET 29
5.2 DATASET PREPROCESSING 29
5.3 BUILDING THE MODEL 31
5.4 SPLITTING THE DATASET AND TRAINING THE MODEL 34
5.5 PREDICTION 34
5.6 TUNING THE PARAMETERS 35

6 RESULT ANALYSIS **37**

7 DISCUSSION **41**

8 CONCLUSION **42**

References **43**

Abstract

Plants are vulnerable to disease due to exposure to various pathogens. Every year, there's a huge yield loss caused by plant disease. Several attempts have been made to reduce the damage caused by it, this model is developed for the identification and prediction of plant diseases based on leaves images. Deep learning is swiftly becoming one of the most essential tools for image classification. This technology is now beginning to be applied to the task related to disease prediction. The model represents research into the main factors that affect the design and effectiveness of a deep convolutional neural network applied to plant diagnostics. The main objective of the model is to present the application of machine learning in plant pathology classification.

1. INTRODUCTION

For a very long time, in the sector of agriculture plant disease had always caused economic and social loss. Multiple measures had been taken for the prevention but not every one of them is effective. Each plant disease has its features and so they all requires different measures. Therefore, developing technologies for precisely monitoring the various crop diseases are extremely crucial in the agricultural management. The famous quote says "a stitch in time saves nine", that is if the prediction of the disease had been done earlier, it'll be much easier to cure the disease with proper medications.

In generic tradition, crop review has been carried out visually by people with some experience and detecting all the ailments in plants. The process of monitoring it manually is not much reliable, many agricultural areas are too widespread to be properly observed in all their extension. More importantly, trained plant pathologists are not easily available, notably in poor and isolated areas.

At present, the potential for automated tools has yet to be apprehended, with the inception of deep learning concepts, the concept of an automated system can be applied here. Deep learning refers to "the use of artificial neural network, architectures that contain a quite large number of processing layers" Among deep learning tools, arguably the most commonly used are the deep neural network. It is an artificial intelligence (AI) function that imitates the workings of the human brain in processing data and creating patterns for use in decision making. Deep learning is a subset of machine learning in artificial intelligence that has networks capable of learning unsupervised from data that is unstructured or unlabelled.

The report provides an in-depth examination of the factors that affect the execution of a deep learning-based machine learning model for plant disease recognition. This analysis provided a wealth of information on the accuracy and methods used. The database, which contains 3624 images is downloaded from an open-source website. The model presents a synopsis of studies that utilize machine learning regarding plant-pathogen interactions and plant disease identifications.

2. RELATED WORK

The University of Alabama, Tuscaloosa has proposed Machine learning in plant disease research paper under European Journal of BioMedical Research. In the agriculture research, machine learning methods are mainly used to detect, identify and predict crop di-seases and plant stress phenotyping. Unlike identification of molecules based on genomics data, machine learning methods in plant diseases research are highly dependent on automated platforms, such as aerial vehicles and ground robots with sensors to collect real-time data from fields. Typically, data were extracted from high-resolution images or sensor data and then preprocessed to remove ir-relevant information. Domain knowledge in plant disease was critical for preprocessing and choosing a proper me-thod, which could significantly improve the performance of machine learning prediction. Machine learning methods were used the most frequently as tools to identify plant diseases. The purpose of such studies are to identify whether or not a disease is present on plants. Several stu-dies have utilized machine learning to detect Huanglong-bing (HLB) for citrus trees. Wetterich et al. detected HLB with SVM based on four features (uniformity, contrast, correlation and homogeneity) extracted from fluorescence imaging spectroscopy. By using this method, the highest classification accuracy was 90 percent for HLB-infected leaves from Brazil but only 61 percent for leaves from USA. La-ter Wetterich et al. used two machine learning algorithms, SVM and artificial neural network (ANN), to discriminate HLB from zinc-deficiency stress on citrus leaves. The ac-curacy was 92.8 percent for SVM and 92.2 percent for ANN respec-tively. Another study showed that SVM with kernel had a better performance in determining the HLB-infected citrus trees than other methods such as linear SVM, linear discriminant analysis and quadratic discriminant analy-sis when using unmanned aerial vehicle (UAV) to collect images of citrus trees . Sankaran et al. developed the detection system for citrus HLB using visible-near in-frared spectroscopy and compared the results of data ana-lysis among LDA, QDA, k-nearest neighbor (KNN) and software independent modeling of classification analogies (SIMCA). QDA and SIMCA-based algorithms had hi-ghest accuracy among all methods.Another important application of machine learning is classification, which classify plant diseases into different stages or different types. The machine learning algorithms were intensively applied on the classification of powdery mildew, a fungal pathogen that causes yield loss in va-rieties of crops. Raza et al. extracted both local and global statistics from thermal and visible light image data and used SVM to identify the powdery mildew-inoculated tomato leaves. They showed that the machine learning system was able to identify the tomato leaves infected na-turally by powdery mildew. An integrated non-super-vised learning algorithms with Bayesian classifiers was presented by Hernandez-Rabadan et al. to separate healthy and infected tomatoes in a uncontrolled environment. Mokhtar et al. reported a machine learning approach using SVM with different kernel functions to differentiate two tomato's viruses and the accuracy was as high as 92 percent based on quadratic kernel function. In addition to powdery mildew, other type of classifications was also im-plemented by machine learning algorithm. In the study by Wahabzada et al., authors proposed methods combining Bayes factors, inter-

pretable matrix factorization and Di-richlet aggregation regression to map three plant disease progress (Pyrenophora teres, Puccinia hordei and Blume-ria graminis hordei). Besides crops, machine learning has also been applied to study interactions between the model plant Arabidopsis and pathogens. Machine learning algorithms such as SVM, Bayesian classifier and random forest were utilized to identify bacterial pathogens with high prediction precisions. By using deep convo-lutional neural networks (CNN) , which is the latest gene-ration of machine learning methods, Sladojevic et al. has successfully recognized 13 different plant pathogens and achieved precision between 91 percent and 98 percent. Further-more, another group also applied deep CNN to classify 14 crop species and 26 diseases with best F1 score of 0.9934 So far, only a few studies have been done to predict the disease development onset and quantification of plant diseases stress. However, prediction and quantification of plant diseases are potentially more im-portant than identification and classification of diseases in the future due to the implications to precision agriculture. Studies of such could lead to preventing crop diseases at early stage and cutting cost for pesticides. In conclusion, in the big data era, machine learning provides a powerful tool to analyze tremendous amount of data. Careful selection of pre-processing data methods and machine learning tools is critical to obtain highest ac-curacy of classification. Meanwhile, compared to traditio-nal methods of identifying genes involved plant pathogen interactions, methods integrating machine learning ap-proaches are relatively scarce in the literature. Thus more machine learning based tools are needed to predict impor-tant plant resistance genes, as well as make contribution to the agriculture. With aerial imaging platforms and sensor technology, collecting field data becomes easier and more precise, which is critical for improving machine learning accuracy. More sofisticated methods such as deep learning algorithms will be applied in detecting plant diseases and discovering plant resistance genes. [12]

3. PRELIMINARIES

3.1 MACHINE LEARNING

Machine Learning is the most popular technique of predicting the future or classifying information to help people in making necessary decisions. Machine Learning is said as a subset of artificial intelligence that is mainly concerned with the development of algorithms which allow a computer to learn from the data and past experiences on their own.

The term machine learning was first introduced by Arthur Samuel in 1959. Machine Learning algorithms are trained over instances or examples through which they learn from past experiences and also analyze the historical data. Therefore, as it trains over the examples, again and again, it is able to identify patterns in order to make predictions about the future. With the help of Machine Learning, we can develop intelligent systems that are capable of taking decisions on an autonomous basis. These algorithms learn from the past instances of data through statistical analysis and pattern matching. Then, based on the learned data, it provides us with the predicted results.

Data is the core backbone of machine learning algorithms. With the help of the historical data, we are able to create more data by training these machine learning algorithms. For example, Generative adversarial networks are an advanced concept of Machine Learning that learns from the historical images through which they are capable of generating more images. This is also applied towards speech and text synthesis. Therefore, Machine Learning has opened up a vast potential for data science applications Machine Learning combines computer science, mathematics, and statistics. Statistics is essential for drawing inferences from the data.

Mathematics is useful for developing machine learning models and finally, computer science is used for implementing algorithms. However, simply building models is not enough. You must also optimize and tune the model appropriately so that it provides you with accurate results. Optimization techniques involve tuning the hyper parameters to reach an optimum result.

Machine Learning is used in every domain. It is being used to impart intelligence to static systems. With the knowledge acquired from the data, it is used to build intelligent products. Machine learning involves computers discovering how they can perform tasks without being explicitly programmed to do so. It involves computers learning from data provided so that they carry out certain tasks. For simple tasks assigned to computers, it is possible to program algorithms telling the machine how to execute all steps required to solve the problem at hand; on the computer's part, no learning is needed. For more advanced tasks, it can be challenging for a human to manually create the needed algorithms.

In practice, it can turn out to be more effective to help the machine develop its own algorithm, rather than having human programmers specify every needed step. Machine learning is closely related to computational statistics, which focuses on making predictions using computers.

3.1.1 NEED

The world today is evolving and so are the needs and requirements of people. Furthermore, we are witnessing a fourth industrial revolution of data. In order to derive meaningful insights from this data and learn from the way in which people and the system interface with the data, we need computational algorithms that can churn the data and provide us with results that would benefit us in various ways.

Machine Learning has revolutionized industries like medicine, healthcare, manufacturing, banking, and several other industries. Therefore, Machine Learning has become an essential part of modern industry. Data is expanding exponentially and in order to harness the power of this data, added by the massive increase in computation power, Machine Learning has added another dimension to the way we perceive information. Machine Learning is being utilized everywhere. The electronic devices you use, the applications that are part of your everyday life are powered by powerful machine learning algorithms.

Machine Learning example – Google is able to provide you with appropriate search results based on browsing habits. Similarly, Netflix is capable of recommending the films or shows that you would want to watch based on the machine learning algorithms that perform predictions based on your watch history.

Furthermore, machine learning has facilitated the automation of redundant tasks that have taken away the need for manual labor. All of this is possible due to the massive amount of data that you generate on a daily basis. Machine Learning facilitates several methodologies to make sense of this data and provide you with steadfast and accurate results.

The need for machine learning is increasing day by day. The reason behind the need for machine learning is that it is capable of doing tasks that are too complex for a person to implement directly. As a human, we have some limitations as we cannot access the huge amount of data manually, so for this, we need some computer systems and here comes the machine learning to make things easy for us. We can train machine learning algorithms by providing them the huge amount of data and let them explore the data, construct the models, and predict the required output automatically.

The performance of the machine learning algorithm depends on the amount of data, and it can be determined by the cost function. With the help of machine learning, we can save both time and money. The importance of machine learning can be easily understood by its uses cases, Currently, machine learning is used in self-driving cars, cyber fraud detection, face recognition, and friend suggestion by Facebook, etc.

Various top companies such as Netflix and Amazon have build machine learning models that are using a vast amount of data to analyze the user interest and recommend product accordingly. In field of machine learning Rapid increment in the production of data and Solving complex problems, which are difficult for a human. Decision making in various sector including finance and Finding hidden patterns and extracting useful information from data.

3.1.2 WORKING

With an exponential increase in data, there is a need for having a system that can handle this massive load of data. Machine Learning models like Deep Learning allow the vast majority of data to be handled with an accurate generation of predictions. Machine Learning has revolutionized the way we perceive information and the various insights we can gain out of it.

Machine learning algorithms use the patterns contained in the training data to perform classification and future predictions. Whenever any new input is introduced to the ML model, it applies its learned patterns over the new data to make future predictions. Based on the final accuracy, one can optimize their models using various standardized approaches. In this way, Machine Learning model learns to adapt to new examples and produce better results.

Suppose we have a complex problem, where we need to perform some predictions, so instead of writing a code for it, we just need to feed the data to generic algorithms, and with the help of these algorithms, machine builds the logic as per the data and predict the output. Machine learning has changed our way of thinking about the problem.

3.1.3 CHARACTERISTICS

- A massive amount of data is being generated by businesses and common people on a regular basis. By visualizing notable relationships in data, businesses can not only make better decisions but build confidence as well. Machine learning offers a number of tools that provide rich snippets of data which can be applied to both unstructured and structured data. With the help of user-friendly automated data visualization platforms in machine learning, businesses can obtain a wealth of new insights in an effort to increase productivity in their processes.

- One of the biggest characteristics of machine learning is its ability to automate repetitive tasks and thus, increasing productivity. A huge number of organizations are already using machine learning-powered paperwork and email automation. In the financial sector, for example, a huge number of repetitive, data-heavy and predictable tasks are needed to be performed. Because of this, this sector uses different types of machine learning solutions to a great extent. The make accounting tasks faster, more insightful, and more accurate. Some aspects that have been already addressed by machine learning include addressing financial queries with the help of chatbots, making predictions, managing expenses, simplifying invoicing, and automating bank reconciliations.

- For any business, one of the most crucial ways to drive engagement, promote brand loyalty and establish long-lasting customer relationships is by triggering meaningful conversations with its target customer base. Machine learning plays a critical role in enabling businesses and brands to spark more valuable conversations in terms of customer engagement. The technology analyzes particular phrases, words, sentences, idioms, and content formats which resonate with certain audience members. You can think of Pinterest which is successfully using machine learning to personalize suggestions to its users. It uses the technology to source content in which users will be interested, based on objects which they have pinned already.

- Machine learning has experienced a great rise in popularity. IoT is being designated as a strategically significant area by many companies. And many others have launched pilot projects to gauge the potential of IoT in the context of business operations. But attaining financial benefits through IoT isn't easy. In order to achieve success, companies, which are offering IoT consulting services and platforms, need to clearly determine the areas that will change with the implementation of IoT strategies. Many of these businesses have failed to address it. In this scenario, machine learning is probably the best technology that

can be used to attain higher levels of efficiency. By merging machine learning with IoT, businesses can boost the efficiency of their entire production processes.

- It's a fact that fostering a positive credit score usually takes discipline, time, and lots of financial planning for a lot of consumers. When it comes to the lenders, the consumer credit score is one of the biggest measures of creditworthiness that involve a number of factors including payment history, total debt, length of credit history etc. But wouldn't it be great if there is a simplified and better measure? With the help of machine learning, lenders can now obtain a more comprehensive consumer picture. They can now predict whether the customer is a low spender or a high spender and understand his/her tipping point of spending. Apart from mortgage lending, financial institutions are using the same techniques for other types of consumer loans.

- Traditionally, data analysis has always been encompassing trial and error method, an approach which becomes impossible when we are working with large and heterogeneous datasets. Machine learning comes as the best solution to all these issues by offering effective alternatives to analyzing massive volumes of data. By developing efficient and fast algorithms, as well as, data-driven models for processing of data in real-time, machine learning is able to generate accurate analysis and results.

- Machine learning characteristics, when merged with big data analytical work, can generate extreme levels of business intelligence with the help of which several different industries are making strategic initiatives. From retail to financial services to healthcare, and many more – machine learning has already become one of the most effective technologies to boost business operations.

- Whether you are convinced or not, the above characteristics of machine learning have contributed heavily toward making it one of the most crucial technology trends – it underlies a huge number of things we use these days without even thinking about them.

3.1.4 APPLICATIONS

- Image recognition is one of the most common applications of machine learning. It is used to identify objects, persons, places, digital images, etc. The popular use case of image recognition and face detection Facebook provides us a feature of auto friend tagging suggestion. Whenever we upload a photo with our Facebook friends, then we automatically get a tagging suggestion with name, and the technology behind this is machine learning's face detection and recognition algorithm.

- While using Google, we get an option of "Search by voice," it comes under speech recognition, and it's a popular application of machine learning.

- Speech recognition is a process of converting voice instructions into text, and it is also known as "Speech to text", or "Computer speech recognition." At present, machine learning algorithms are widely used by various applications of speech recognition. Google assistant, Siri, Cortana, and Alexa are using speech recognition technology to follow the voice instructions.

- If we want to visit a new place, we take help of Google Maps, which shows us the correct path with the shortest route and predicts the traffic conditions. Everyone who is using Google Map is helping this app to make it better. It takes information from the user and sends back to its database to improve the performance.

- It predicts the traffic conditions such as whether traffic is cleared, slow-moving, or heavily congested with the help of two ways:

    1. Real Time location of the vehicle form Google Map app and sensors
    2. Average time has taken on past days at the same time.

- Machine learning is widely used by various e-commerce and entertainment companies such as Amazon, Netflix, etc., for product recommendation to the user. Whenever we search for some product on Amazon, then we started getting an advertisement for the same product while internet surfing on the same browser and this is because of machine learning. Google understands the user interest using various machine learning algorithms and suggests the product as per customer interest. As similar, when we use Netflix, we find some recommendations for entertainment series, movies, etc., and this is also done with the help of machine learning.

- One of the most exciting applications of machine learning is self-driving cars. Machine learning plays a significant role in self-driving cars. Tesla, the most popular car manufacturing company is working on self-driving car. It is using unsupervised learning method to train the car models to detect people and objects while driving.

- Whenever we receive a new email, it is filtered automatically as important, normal, and spam. We always receive an important mail in our inbox with the important symbol and spam emails in our spam box, and the technology behind this is Machine learning. Some machine learning algorithms such as Multi-Layer Perception, Decision tree, and Naive Bayes classifier are used for email spam filtering and malware detection.

- We have various virtual personal assistants such as Google assistant, Alexa, Cortana, Siri. As the name suggests, they help us in finding the information using our voice instruction. These assistants can help us in various ways just by our voice instructions such as Play music, call someone, Open an email, Scheduling an appointment, etc. These virtual assistants use machine learning algorithms as an important part. These assistant record our voice instructions, send it over the server on a cloud, and decode it using ML algorithms and act accordingly.

- Machine learning is making our online transaction safe and secure by detecting fraud transaction. Whenever we perform some online transaction, there may be various ways that a fraudulent transaction can take place such as fake accounts, fake ids, and steal money in the middle of a transaction. So to detect this, Feed Forward Neural network helps us by checking whether it is a genuine transaction or a fraud transaction. For each genuine transaction, the output is converted into some hash values, and these values become the input for the next round. For each genuine transaction, there is a specific pattern which gets change for the fraud transaction hence, it detects it and makes our online transactions more secure.

- Machine learning is widely used in stock market trading. In the stock market, there is always a risk of up and downs in shares, so for this machine learning's long short term memory neural network is used for the prediction of stock market trends.

- In medical science, machine learning is used for diseases diagnoses. With this, medical technology is growing very fast and able to build 3D models that can predict the exact position of lesions in the brain. It helps in finding brain tumors and other brain-related diseases easily.

- Nowadays, if we visit a new place and we are not aware of the language then it is not a problem at all, as for this also machine learning helps us by converting the text into our known languages. Google's GNMT (Google Neural Machine Translation) provide this feature, which is a Neural Machine Learning that translates the text into our familiar language, and it called as automatic translation. The technology behind the automatic translation is a sequence to sequence learning algorithm, which is used with image recognition and translates the text from one language to another language.

3.1.5 CLASSIFICATION

Supervised Machine Learning

In Supervised Learning, the dataset on which we train our model is labeled. There is a clear and distinct mapping of input and output. Based on the example inputs, the model is able to get trained in the instances. An example of supervised learning is spam filtering. Based on the labeled data, the model is able to determine if the data is spam or ham. This is an easier form of training. Spam filtering is an example of this type of machine learning algorithm.

Unsupervised Machine Learning

In Unsupervised Learning, there is no labeled data. The algorithm identifies the patterns within the dataset and learns them. The algorithm groups the data into various clusters based on their density. Using it, one can perform visualization on high dimensional data. One example of this type of Machine learning algorithm is the Principle Component Analysis. Furthermore, K-Means Clustering is another type of Unsupervised Learning where the data is clustered in groups of a similar order.

Reinforcement Machine Learning

Reinforcement Learning is an emerging and most popular type of Machine Learning Algorithm. It is used in various autonomous systems like cars and industrial robotics. The aim of this algorithm is to reach a goal in a dynamic environment. It can reach this goal based on several rewards that are provided to it by the system.

It is most heavily used in programming robots to perform autonomous actions. It is also used in making intelligent self-driving cars. Let us consider the case of robotic navigation. Furthermore, the efficiency can be improved with further experimentation with the agent in its environment. This is the main principle behind reinforcement learning.

There are similar sequences of action in a reinforcement learning model.

Input: The input should be an initial state from which the model will start

Output: There are many possible output as there are variety of solution to a particular problem

Training: The training is based upon the input, The model will return a state and the user will decide to reward or punish the model based on its output. The model keeps continues to learn. The best solution is decided based on the maximum reward. Reinforcement learning is all about making decisions sequentially.

In simple words we can say that the output depends on the state of the current input and the next input depends on the output of the previous input. In Reinforcement learning decision is dependent, So we give labels to sequences of dependent decisions.

3.1.6 SUPERVISED MACHINE LEARNING

3.1.6.1 Regression

Regression algorithms are used if there is a relationship between the input variable and the output variable. It is used for the prediction of continuous variables, such as Weather forecasting, Market Trends, etc. Below are some popular Regression algorithms which come under supervised learning:

Linear Regression

Linear regression is one of the easiest and most popular Machine Learning algorithms. It is a statistical method that is used for predictive analysis. Linear regression makes predictions for continuous/real or numeric variables such as sales, salary, age, product price, etc. Linear regression algorithm shows a linear relationship between a dependent (y) and one or more independent (y) variables, hence called as linear regression. Since linear regression shows the linear relationship, which means it finds how the value of the dependent variable is changing according to the value of the independent variable.

Types of linear regression:

- **Simple Linear Regression**: If a single independent variable is used to predict the value of a numerical dependent variable, then such a Linear Regression algorithm is called Simple Linear Regression.
- **Multiple Linear Regression**: If more than one independent variable is used to predict the value of a numerical dependent variable, then such a Linear Regression algorithm is called Multiple Linear Regression.

Bayesian Regression

In the Bayesian viewpoint, we formulate linear regression using probability distributions rather than point estimates. The response, y, is not estimated as a single value, but is assumed to be drawn from a probability distribution.

The output, y is generated from a normal (Gaussian) Distribution characterized by a mean and variance. The mean for linear regression is the transpose of the weight matrix multiplied by the predictor matrix. The variance is the square of the standard deviation (multiplied by the Identity matrix because this is a multi-dimensional formulation of the model).

The aim of Bayesian Linear Regression is not to find the single "best" value of the model parameters, but rather to determine the posterior distribution for the model parameters. Not only

is the response generated from a probability distribution, but the model parameters are assumed to come from a distribution as well.

Polynomial Regression

Polynomial Regression is a regression algorithm that models the relationship between a dependent(y) and independent variable(x) as nth degree polynomial.

It is also called the special case of Multiple Linear Regression in ML. Because we add some polynomial terms to the Multiple Linear regression equation to convert it into Polynomial Regression. It is a linear model with some modification in order to increase the accuracy. The dataset used in Polynomial regression for training is of non-linear nature. It makes use of a linear regression model to fit the complicated and non-linear functions and datasets. Hence, "In Polynomial regression, the original features are converted into Polynomial features of required degree (2,3,..,n) and then modeled using a linear model." If we apply a linear model on a linear dataset, then it provides us a good result as we have seen in Simple Linear Regression, but if we apply the same model without any modification on a non-linear dataset, then it will produce a drastic output. Due to which loss function will increase, the error rate will be high, and accuracy will be decreased. Hence, if the datasets are arranged in a non-linear fashion, then we should use the Polynomial Regression model instead of Simple Linear Regression.

Non-linear Regression

Nonlinear regression is a statistical technique that helps describe nonlinear relationships in experimental data. Nonlinear regression models are generally assumed to be parametric, where the model is described as a nonlinear equation. Typically machine learning methods are used for non-parametric nonlinear regression. Parametric nonlinear regression models the dependent variable (also called the response) as a function of a combination of nonlinear parameters and one or more independent variables (called predictors). The model can be univariate (single response variable) or multivariate (multiple response variables). The parameters can take the form of an exponential, trigonometric, power, or any other nonlinear function.

3.1.6.2 Classification

Random Forest

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model. "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."

Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output. "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." Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output. There should be some actual values in the feature variable of the dataset so that the classifier can predict accurate results rather than a guessed result.

The predictions from each tree must have very low correlations. 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. Suppose there is a dataset that contains multiple fruit images. So, this dataset is given to the Random forest classifier. The dataset is divided into subsets and given to each decision tree. During the training phase, each decision tree produces a prediction result, and when a new data point occurs, then based on the majority of results, the Random Forest classifier predicts the final decision.

Random Forest is capable of performing both Classification and Regression tasks. It is capable of handling large datasets with high dimensionality. It enhances the accuracy of the model and prevents the overfitting issue. It overcomes the problem of overfitting by averaging or combining the results of different decision trees. Random forests work well for a large range of data items than a single decision tree does. Random Forest algorithms maintains good accuracy even a large proportion of the data is missing.

Decision Tree Classifier

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 subtrees. There are various algorithms in Machine learning, so choosing the best algorithm for the given dataset and problem is the main point to remember while creating a machine learning model. Decision Trees usually mimic human thinking ability while making a decision, so it is easy to understand. The logic behind the decision tree can be easily understood because it shows a tree-like structure.

Root Node: Root node is from where the decision tree starts. It represents the entire dataset, which further gets divided into two or more homogeneous sets.

Leaf Node: Leaf nodes are the final output node, and the tree cannot be segregated further after getting a leaf node.

Splitting: Splitting is the process of dividing the decision node/root node into sub-nodes according to the given conditions.

Branch/Sub Tree: A tree formed by splitting the tree.

Pruning: Pruning is the process of removing the unwanted branches from the tree.

Parent/Child node: The root node of the tree is called the parent node, and other nodes are called the child nodes.

In a decision tree, for predicting the class of the given dataset, the algorithm starts from the

root node of the tree. This algorithm compares the values of root attribute with the record (real dataset) attribute and, based on the comparison, follows the branch and jumps to the next node.

For the next node, the algorithm again compares the attribute value with the other sub-nodes and move further. It continues the process until it reaches the leaf node of the tree. The complete process can be better understood using the below algorithm:

Step-1: Begin the tree with the root node, says S, which contains the complete dataset.

Step-2: Find the best attribute in the dataset using Attribute Selection Measure (ASM).

Step-3: Divide the S into subsets that contains possible values for the best attributes.

Step-4: Generate the decision tree node, which contains the best attribute.

Step-5: Recursively make new decision trees using the subsets of the dataset created in step-3. Continue this process until a stage is reached where you cannot further classify the nodes and called the final node as a leaf node.

Suppose there is a candidate who has a job offer and wants to decide whether he should accept the offer or Not. So, to solve this problem, the decision tree starts with the root node (Salary attribute by ASM). The root node splits further into the next decision node (distance from the office) and one leaf node based on the corresponding labels. The next decision node further gets split into one decision node (Cab facility) and one leaf node. Finally, the decision node splits into two leaf nodes (Accepted offers and Declined offer).

Logistic Regression Algorithm

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.

Support Vector Machine Algorithm

Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning. The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.

SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine.

SVM can be of two types:

- Linear SVM: Linear SVM is used for linearly separable data, which means if a dataset can be classified into two classes by using a single straight line, then such data is termed as linearly separable data, and classifier is used called as Linear SVM classifier.
- Non-linear SVM: Non-Linear SVM is used for non-linearly separated data, which means if a dataset cannot be classified by using a straight line, then such data is termed as non-linear data and classifier used is called as Non-linear SVM classifier.

3.1.7 UNSUPERVISED MACHINE LEARNING

unsupervised learning is a machine learning technique in which models are not supervised using training dataset. Instead, models itself find the hidden patterns and insights from the given data. It can be compared to learning which takes place in the human brain while learning new things.

Types of unsupervised learning algorithm:

3.1.7.1 Clustering

Clustering is a method of grouping the objects into clusters such that objects with most similarities remains into a group and has less or no similarities with the objects of another group. Cluster analysis finds the commonalities between the data objects and categorizes them as per the presence and absence of those commonalities. It is basically a type of unsupervised learning method . An unsupervised learning method is a method in which we draw references from datasets consisting of input data without labelled responses. Generally, it is used as a process to find meaningful structure, explanatory underlying processes, generative features, and groupings inherent in a set of examples.

Clustering is the task of dividing the population or data points into a number of groups such that data points in the same groups are more similar to other data points in the same group and dissimilar to the data points in other groups. It is basically a collection of objects on the basis of similarity and dissimilarity between them. Clustering is very much important as it determines the intrinsic grouping among the unlabeled data present. There are no criteria for a good clustering. It depends on the user, what is the criteria they may use which satisfy their need. For instance, we could be interested in finding representatives for homogeneous groups (data reduction), in finding "natural clusters" and describe their unknown properties ("natural" data types), in finding useful and suitable groupings ("useful" data classes) or in finding unusual data objects (outlier detection). This algorithm must make some assumptions which constitute the similarity of points and each assumption make different and equally valid clusters.

k-means clustering

K-Means Clustering is an unsupervised learning algorithm that is used to solve the clustering problems in machine learning or data science. In this topic, we will learn what is K-means

clustering algorithm, how the algorithm works, along with the Python implementation of k-means clustering. K-Means Clustering is an Unsupervised Learning algorithm, which groups the unlabeled dataset into different clusters.

Here K defines the number of pre-defined clusters that need to be created in the process, as if K=2, there will be two clusters, and for K=3, there will be three clusters, and so on. It allows us to cluster the data into different groups and a convenient way to discover the categories of groups in the unlabeled dataset on its own without the need for any training. It is a centroid-based algorithm, where each cluster is associated with a centroid. The main aim of this algorithm is to minimize the sum of distances between the data point and their corresponding clusters. The algorithm takes the unlabeled dataset as input, divides the dataset into k-number of clusters, and repeats the process until it does not find the best clusters. The value of k should be predetermined in this algorithm.

The k-means clustering algorithm mainly performs two tasks:

- Determines the best value for K center points or centroids by an iterative process.
- Assigns each data point to its closest k-center. Those data points which are near to the particular k-center, create a cluster.

The working of the K-Means algorithm is explained in the below steps:

Step-1: Select the number K to decide the number of clusters.

Step-2: Select random K points or centroids. (It can be other from the input dataset).

Step-3: Assign each data point to their closest centroid, which will form the predefined K clusters.

Step-4: Calculate the variance and place a new centroid of each cluster.

Step-5: Repeat the third steps, which means reassign each datapoint to the new closest centroid of each cluster.

Step-6: If any reassignment occurs, then go to step-4 else go to FINISH.

Step-7: The model is ready.

KNN algorithm

K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique. K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories. K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm. K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems. K-NN is a non-parametric algorithm, which means it does not make any assumption on underlying data. It is also called a lazy learner algorithm because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data.

The K-NN working can be explained on the basis of the below algorithm:

Step-1: Select the number K of the neighbors

Step-2: Calculate the Euclidean distance of K number of neighbors

Step-3: Take the K nearest neighbors as per the calculated Euclidean distance.

Step-4: Among these k neighbors, count the number of the data points in each category.

Step-5: Assign the new data points to that category for which the number of the neighbor is maximum.

Step-6: Our model is ready.

Hierarchical Clustering

Hierarchical clustering is another unsupervised machine learning algorithm, which is used to group the unlabeled datasets into a cluster and also known as hierarchical cluster analysis or HCA. In this algorithm, we develop the hierarchy of clusters in the form of a tree, and this tree-shaped structure is known as the dendrogram. Sometimes the results of K-means clustering and hierarchical clustering may look similar, but they both differ depending on how they work. As there is no requirement to predetermine the number of clusters as we did in the K-Means algorithm. The hierarchical clustering technique has two approaches:

- Agglomerative: Agglomerative is a bottom-up approach, in which the algorithm starts with taking all data points as single clusters and merging them until one cluster is left.
- Divisive: Divisive algorithm is the reverse of the agglomerative algorithm as it is a top-down approach.

3.1.7.2 Association

An association rule is an unsupervised learning method which is used for finding the relationships between variables in the large database. It determines the set of items that occurs together in the dataset. Association rule makes marketing strategy more effective. Such as people who buy X item (suppose a bread) are also tend to purchase Y (Butter/Jam) item. A typical example of Association rule is Market Basket Analysis. Association rule mining finds interesting associations and relationships among large sets of data items. This rule shows how frequently a itemset occurs in a transaction. A typical example is Market Based Analysis.

Market Based Analysis is one of the key techniques used by large relations to show associations between items. It allows retailers to identify relationships between the items that people buy together frequently.

3.1.8 REINFORCEMENT MACHINE LEARNING

There are two types of Reinforcement:

3.1.8.1 Positive Reinforcement

Positive Reinforcement is defined as when an event, occurs due to a particular behavior, increases the strength and the frequency of the behavior. In other words, it has a positive effect on behavior. Advantages are:

- Maximizes Performance
- Sustain Change for a long period of time
- Disadvantages of reinforcement learning:
- Too much Reinforcement can lead to overload of states which can diminish the results.

3.1.8.2 Negative Reinforcement

It is defined as strengthening of a behavior because a negative condition is stopped or avoided. Advantages are:

- Increases Behavior
- Provide defiance to minimum standard of performance

[6]

3.2 DEEP LEARNING

Deep learning is based on the branch of machine learning, which is a subset of artificial intelligence. Since neural networks imitate the human brain and so deep learning will do. In deep learning, nothing is programmed explicitly. Basically, it is a machine learning class that makes use of numerous nonlinear processing units so as to perform feature extraction as well as transformation. The output from each preceding layer is taken as input by each one of the successive layers. Deep learning models are capable enough to focus on the accurate features themselves by requiring a little guidance from the programmer and are very helpful in solving out the problem of dimensionality. Deep learning algorithms are used, especially when we have a huge no of inputs and outputs. Since deep learning has been evolved by the machine learning, which itself is a subset of artificial intelligence and as the idea behind the artificial intelligence is to mimic the human behavior, so same is "the idea of deep learning to build such algorithm that can mimic the brain". Deep learning is implemented with the help of Neural Networks, and the idea behind the motivation of Neural Network is the biological neurons, which is nothing but a brain cell. [2]

3.2.1 APPLICATIONS

- Self-Driving Cars

In self-driven cars, it is able to capture the images around it by processing a huge amount of data, and then it will decide which actions should be incorporated to take a left or right or should it stop. So, accordingly, it will decide what actions it should take, which will further reduce the accidents that happen every year.

- Voice Controlled Assistance

When we talk about voice control assistance, then Siri is the one thing that comes into our mind. So, you can tell Siri whatever you want it to do it for you, and it will search it for you and display it for you.

- Automatic Image Caption Generation

Whatever image that you upload, the algorithm will work in such a way that it will generate caption accordingly. If you say blue colored eye, it will display a blue-colored eye with a caption at the bottom of the image.

- Automatic Machine Translation

With the help of automatic machine translation, we are able to convert one language into another with the help of deep learning.[2]

3.2.2 NEURAL NETWORK

A neural network is a series of algorithms that endeavors to recognize underlying relationships in a set of data through a process that mimics the way the human brain operates. In this sense, neural networks refer to systems of neurons, either organic or artificial in nature. Neural networks can adapt to changing input; so the network generates the best possible result without needing to redesign the output criteria. The concept of neural networks, which has its roots in artificial intelligence, is swiftly gaining popularity in the development of trading systems. Neural networks, in the world of finance, assist in the development of such process as time-series forecasting, algorithmic trading, securities classification, credit risk modeling and constructing proprietary indicators and price derivatives. A neural network works similarly to the human brain's neural network. A "neuron" in a neural network is a mathematical function that collects and classifies information according to a specific architecture. The network bears a strong resemblance to statistical methods such as curve fitting and regression analysis. A neural network contains layers of interconnected nodes. Each node is a perceptron and is similar to a multiple linear regression. The perceptron feeds the signal produced by a multiple linear regression into an activation function that may be nonlinear. In a multi-layered perceptron (MLP), perceptrons are arranged in interconnected layers. The input layer collects input patterns. The output layer has classifications or output signals to which input patterns may map. For instance, the patterns may comprise a list of quantities for technical indicators about a security; potential outputs could be "buy," "hold" or "sell." Hidden layers fine-tune the input weightings until the neural network's margin of error is minimal. It is hypothesized that hidden layers extrapolate salient features in the input data that have predictive power regarding the outputs. This describes feature extraction, which accomplishes a utility similar to statistical techniques such as principal component analysis. [3]

3.2.2.1 Application

Neural networks are broadly used, with applications for financial operations, enterprise planning, trading, business analytics and product maintenance. Neural networks have also gained

widespread adoption in business applications such as forecasting and marketing research solutions, fraud detection and risk assessment. A neural network evaluates price data and unearths opportunities for making trade decisions based on the data analysis. The networks can distinguish subtle nonlinear interdependencies and patterns other methods of technical analysis cannot. According to research, the accuracy of neural networks in making price predictions for stocks differs. Some models predict the correct stock prices 50 to 60 percent of the time while others are accurate in 70 percent of all instances. Some have posited that a 10 percent improvement in efficiency is all an investor can ask for from a neural network. There will always be data sets and task classes that a better analyzed by using previously developed algorithms. It is not so much the algorithm that matters; it is the well-prepared input data on the targeted indicator that ultimately determines the level of success of a neural network.[3]

3.2.3 CONVOLUTIONAL NEURAL NETWORK

A Convolutional Neural Network is a Deep Learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other. The pre-processing required in a ConvNet is much lower as compared to other classification algorithms. While in primitive methods filters are hand-engineered, with enough training, ConvNets have the ability to learn these filters/characteristics. The architecture of a ConvNet is analogous to that of the connectivity pattern of Neurons in the Human Brain and was inspired by the organization of the Visual Cortex. Individual neurons respond to stimuli only in a restricted region of the visual field known as the Receptive Field. A collection of such field overlap to cover the entire visual area. [5]

3.2.3.1 why convnets over feed-forward neural nets

An image is nothing but a matrix of pixel values, right? So why not just flatten the image and feed it to a Multi-Level Perceptron for classification purposes? Uh.. not really. In cases of extremely basic binary images, the method might show an average precision score while performing prediction of classes but would have little to no accuracy when it comes to complex images having pixel dependencies throughout. A ConvNet is able to successfully capture the Spatial and Temporal dependencies in an image through the application of relevant filters. The architecture performs a better fitting to the image dataset due to the reduction in the number of parameters involved and reusability of weights. In other words, the network can be trained to understand the sophistication of the image better.[11]

3.2.3.2 input image

In the figure, we have an RGB image which has been separated by its three color planes — Red, Green, and Blue. There are a number of such color spaces in which images exist — Grayscale, RGB, HSV, CMYK, etc. The role of the ConvNet is to reduce the images into a form which is easier to process, without losing features which are critical for getting a good prediction. This is important when we are to design an architecture which is not only good at learning features but also is scalable to massive datasets. [11]

3.2.4 RECURRENT NEURAL NETWORK

Recurrent Neural Networks (RNNs) add an interesting twist to basic neural networks. A vanilla neural network takes in a fixed size vector as input which limits its usage in situations that involve a 'series' type input with no predetermined size. Recurrent Neural Network remembers the past and it's decisions are influenced by what it has learnt from the past. Note: Basic feed forward networks "remember" things too, but they remember things they learnt during training. For example, an image classifier learns what a "1" looks like during training and then uses that knowledge to classify things in production. While RNNs learn similarly while training, in addition, they remember things learnt from prior input(s) while generating output(s). It's part of the network. RNNs can take one or more input vectors and produce one or more output vectors and the output(s) are influenced not just by weights applied on inputs like a regular NN, but also by a "hidden" state vector representing the context based on prior input(s)/output(s). So, the same input could produce a different output depending on previous inputs in the series. In summary, in a vanilla neural network, a fixed size input vector is transformed into a fixed size output vector. Such a network becomes "recurrent" when you repeatedly apply the transformations to a series of given input and produce a series of output vectors. There is no pre-set limitation to the size of the vector. And, in addition to generating the output which is a function of the input and hidden state, we update the hidden sate itself based on the input and use it in processing the next input. Multiple different weights are applied to the different parts of an input item generating a hidden layer neuron, which in turn is transformed using further weights to produce an output. There seems to be a lot of weights in play here. Applying the same weights over and over again to different items in the input series. Figure deals with "a" single input whereas the second figure represents multiple inputs from a series. But nevertheless, intuitively speaking, as the number of inputs increase, shouldn't the number of weights in play increase as well sharing parameters across inputs If don't share parameters across inputs, then it becomes like a vanilla neural network where each input node requires weights of their own. This introduces the constraint that the length of the input has to be fixed and that makes it impossible to leverage a series type input where the lengths differ and is not always known. [13]

4. PREPROCESSING TECHNIQUES REQUIRED

Image preprocessing is divided into analogue image processing and digital image processing.

Digital image processing is the use of computer algorithms to perform image processing on digital images. As a subfield of digital signal processing, digital image processing has many advantages over analogue image processing. It allows a much wider range of algorithms to be applied to the input data — the aim of digital image processing is to improve the image data (features) by suppressing unwanted distortions and/or enhancement of some important image features so that our AI-Computer Vision models can benefit from this improved data to work on.

An image is nothing more than a two-dimensional array of numbers(or pixels) ranging between 0 and 255. It is defined by the mathematical function f(x,y) where x and y are the two co-ordinates horizontally and vertically.

The value of f(x,y) at any point is giving the pixel value at that point of an image. [9]

Python provides lots of libraries for image processing, including

- OpenCV Image processing library mainly focused on real-time computer vision with application in wide-range of areas like 2D and 3D feature toolkits, facial and gesture recognition, Human-computer interaction, Mobile robotics, Object identification and others.
- Numpy and Scipy libraries For image manipuation and processing.
- Sckikit Provides lots of alogrithms for image processing.
- Python Imaging Library (PIL) To perform basic operations on images like create thumnails, resize, rotation, convert between different file formats etc.

Using, the above libraries, the image preprocessing steps are carried out. The steps to be taken for image preprocessing are :

- Read image
- Resize image
- Remove noise(Denoise)
- Segmentation
- Morphology(smoothing edges)

Now, let's discuss each of them in detail.

4.1 READING IMAGES

In this step, we store the path to our image data set into a variable then we create a function to load folders containing images into arrays.

For example,

```
import os
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
import cv2

imagepath = "Path to your dataset"

def loadImages(path):
imagefiles = sorted([os.path.join(path, 'train', file)
for file in os.listdir(path + "/train") if
file.endswith('.png')])
return imagefiles
```

Just like that, we stored the folder containing training images from the Dataset into the array image files. [1]

4.2 RESIZING IMAGE

In this step in order to visualize the change, we are going to create two functions to display the images the first being a one to display one image and the second for two images. After that, we then create a function called processing that just receives the images as a parameter.

Need for resizing image

Some images captured by a camera and fed to our AI algorithm vary in size, therefore, we should establish a base size for all images fed into our AI algorithms.

For example,

```
def processing(data):
img = [cv2.imread(i, cv2.IMREADUNCHANGED) for i in data[:3]]
print('Original size',img[0].shape)
height = 200
width = 200
dim = (width, height)
```

```
resimg = []
for i in range(len(img)):
res = cv2.resize(img[i], dim,
interpolation=cv2.INTERLINEAR)
resimg.append(res)
```

This code would be responsible for converting any size of image to the predefined size that in this case is 200 in height and 200 in width. [8]

4.3 DENOISING IMAGE

One of the fundamental challenges in the field of image processing and computer vision is image denoising, where the underlying goal is to estimate the original image by suppressing noise from a noise-contaminated version of the image. Image noise may be caused by different intrinsic (i.e., sensor) and extrinsic (i.e., environment) conditions which are often not possible to avoid in practical situations. Therefore, image denoising plays an important role in a wide range of applications such as image restoration, visual tracking, image registration, image segmentation, and image classification, where obtaining the original image content is crucial for strong performance.

While many algorithms have been proposed for the purpose of image denoising, the problem of image noise suppression remains an open challenge, especially in situations where the images are acquired under poor conditions where the noise level is very high.

Still, inside the function Processing() we add this code to smooth our image to remove unwanted noise. We do this using gaussian blur. Gaussian blur (also known as Gaussian smoothing) is the result of blurring an image by a Gaussian function. It is a widely used effect in graphics software, typically to reduce image noise. The visual effect of this blurring technique is a smooth blur resembling that of viewing the image through a translucent screen, distinctly different from the bokeh effect produced by an out-of-focus lens or the shadow of an object under usual illumination. Gaussian smoothing is also used as a pre-processing stage in computer vision algorithms in order to enhance image structures at different scales.

For example,

```
nonoise = []
for i in range(len(resimg)):
blur = cv2.GaussianBlur(resimg[i], (5, 5), 0)
nonoise.append(blur)

image = nonoise[1]
display(original, image, 'Original', 'Blured')
```

Now, in the above techniques, we took a small neighbourhood around a pixel and did some operations like gaussian weighted average, median of the values etc to replace the central element. In short, noise removal at a pixel was local to its neighbourhood.

There is a property of noise. Noise is generally considered to be a random variable with zero mean. Consider a noisy pixel, p = p0 + n where p0 is the true value of pixel and n is the noise in that pixel. You can take large number of same pixels (say N) from different images and computes their average. Ideally, you should get p = p0 since mean of noise is zero.

You can verify it yourself by a simple setup. Hold a static camera to a certain location for a couple of seconds. This will give you plenty of frames, or a lot of images of the same scene. Then write a piece of code to find the average of all the frames in the video (This should be too simple for you now). Compare the final result and first frame. You can see reduction in noise. Unfortunately this simple method is not robust to camera and scene motions. Also often there is only one noisy image available.

So idea is simple, we need a set of similar images to average out the noise. Here, the idea of non local means denoising comes into play.

Non-local means is an algorithm in image processing for image denoising. Unlike "local mean" filters, which take the mean value of a group of pixels surrounding a target pixel to smooth the image, non-local means filtering takes a mean of all pixels in the image, weighted by how similar these pixels are to the target pixel. This results in much greater post-filtering clarity, and less loss of detail in the image compared with local mean algorithms.

If compared with other well-known denoising techniques, non-local means adds "method noise" (i.e. error in the denoising process) which looks more like white noise, which is desirable because it is typically less disturbing in the denoised product. Recently non-local means has been extended to other image processing applications such as deinterlacing, view interpolation, and depth maps regularization.

OpenCV provides four variations of this technique.

- **cv.fastNlMeansDenoising()** - works with a single grayscale images
- **cv.fastNlMeansDenoisingColored()** - works with a color image.
- **cv.fastNlMeansDenoisingMulti()** - works with image sequence captured in short period of time (grayscale images)
- **cv.fastNlMeansDenoisingColoredMulti()** - same as above, but for color images.

Common arguments are:

- h : parameter deciding filter strength. Higher h value removes noise better, but removes details of image also. (10 is ok)
- hForColorComponents : same as h, but for color images only. (normally same as h)
- templateWindowSize : should be odd. (recommended 7)
- searchWindowSize : should be odd. (recommended 21)

For example, using the fastNlMeansDenoisingColored function

```
import numpy as np
import cv2 as cv
from matplotlib import pyplot as plt
img = cv.imread('die.png')
dst = cv.fastNlMeansDenoisingColored(img,None,10,10,7,21) [7]
```

4.4 SEGMENTATION AND MORPHOLOGY

In digital image processing and computer vision, image segmentation is the process of partitioning a digital image into multiple segments (sets of pixels, also known as image objects). The goal of segmentation is to simplify and/or change the representation of an image into something that is more meaningful and easier to analyze. Image segmentation is typically used to locate objects and boundaries (lines, curves, etc.) in images. More precisely, image segmentation is the process of assigning a label to every pixel in an image such that pixels with the same label share certain characteristics.

The result of image segmentation is a set of segments that collectively cover the entire image, or a set of contours extracted from the image (see edge detection). Each of the pixels in a region are similar with respect to some characteristic or computed property, such as color, intensity, or texture. Adjacent regions are significantly different with respect to the same characteristic(s). When applied to a stack of images, typical in medical imaging, the resulting contours after image segmentation can be used to create 3D reconstructions with the help of interpolation algorithms like marching cubes.

In this step, we step we are going to segment the image, separating the background from foreground objects and we are going to further improve our segmentation with more noise removal.

For example,

```
gray = cv2.cvtColor(image, cv2.COLORRGB2GRAY)
ret, thresh = cv2.threshold(gray, 0, 255, cv2.THRESHBINARYINV + cv2.THRESHOTSU)

display(original, thresh, 'Original', 'Segmented')
```

Morphological image processing is a collection of non-linear operations related to the shape or morphology of features in an image. According to Wikipedia, morphological operations rely only on the relative ordering of pixel values, not on their numerical values, and therefore are especially suited to the processing of binary images. Morphological operations can also be applied to greyscale images such that their light transfer functions are unknown and therefore their absolute pixel values are of no or minor interest.

Morphological techniques probe an image with a small shape or template called a structuring element. The structuring element is positioned at all possible locations in the image and it is compared with the corresponding neighbourhood of pixels. [4]

We can see that the image above needs further enhancement, therefore, we apply another blur to improve the looks with the following code:

```
kernel = np.ones((3, 3), np.uint8)
opening = cv2.morphologyEx(thresh, cv2.MORPHOPEN, kernel, iterations=2)

surebg = cv2.dilate(opening, kernel, iterations=3)

disttransform = cv2.distanceTransform(opening, cv2.DISTL2, 5)
ret, surefg = cv2.threshold(disttransform, 0.7 * disttransform.max(), 255, 0)

surefg = np.uint8(surefg)
unknown = cv2.subtract(surebg, surefg)

display(original, surebg, 'Original', 'Segmented Background')
```

4.5 EDGE DETECTION

Edge detection includes a variety of mathematical methods that aim at identifying points in a digital image at which the image brightness changes sharply or, more formally, has discontinuities. The points at which image brightness changes sharply are typically organized into a set of curved line segments termed edges. The same problem of finding discontinuities in one-dimensional signals is known as step detection and the problem of finding signal discontinuities over time is known as change detection. Edge detection is a fundamental tool in image processing, machine vision and computer vision, particularly in the areas of feature detection and feature extraction.

Canny Edge Detection is a popular edge detection algorithm. It was developed by John F. Canny in 1986. It is a multi-stage algorithm. Its stages are as follows:

- Noise Reduction
- Finding intensity gradient of the image
- Non-maximum Suppression
- Hysteresis Thresholding

OpenCV puts all the above in single function, cv2.Canny(). We will see how to use it. First argument is our input image. Second and third arguments are our minVal and maxVal respectively. Third argument is aperture size. It is the size of Sobel kernel used for find image gradients. By default it is 3. Last argument is L2gradient which specifies the equation for finding gradient magnitude. [10]

For example,

```
import cv2
import numpy as np
from matplotlib import pyplot as plt

img = cv2.imread('messi5.jpg',0)
edges = cv2.Canny(img,100,200)
```

5. PROPOSED METHODOLOGY

5.1 DATASET

The dataset is downloaded from www.kaggle.com. The dataset is a problem of multiclass classification. In machine learning, multiclass or multinomial classification is the problem of classifying instances into one of three or more classes (classifying instances into one of two classes is called binary classification). The dataset contains 3642 images named with train and test purposes. There are 1820 images for the train and the same number of images for under category of test. The dataset file has five categories, imageId, Healthy, multiplueDiseases, rust, and scab. These categories put this dataset in multiclass classification. There are two files "train.csv" and "test.csv" but the "test.csv" file is empty so we relied on splitting the training file to get the accuracy for testing.

DESCRIPTION:

Misdiagnosis of the many diseases impacting agricultural crops can lead to misuse of chemicals leading to the emergence of resistant pathogen strains, increased input costs, and more outbreaks with significant economic loss and environmental impacts. Current disease diagnosis based on human scouting is time-consuming and expensive, and although computer-vision based models have the promise to increase efficiency, the great variance in symptoms due to age of infected tissues, genetic variations, and light conditions within trees decreases the accuracy of detection.

Objectives of 'Plant Pathology Challenge' are to train a model using images of training dataset to

- Accurately classify a given image from testing dataset into different diseased category or a healthy leaf.
- Accurately distinguish between many diseases, sometimes more than one on a single leaf
- Deal with rare classes and novel symptoms
- Address depth perception—angle light, shade, physiological age of the leaf
- Incorporate expert knowledge in identification, annotation, quantification, and guiding computer vision to search for relevant features during learning.

5.2 DATASET PREPROCESSING

For data preprocessing, the imported modules are:

- import numpy as np
- import pandas as pd
- import matplotlib.pyplot as plt
- import os
- import cv2
- import keras

IMAGE PREPROCESSING:

print(image)
image1 = cv2.imread(image)
image2 = cv2.resize(image1, (200, 200))
image3 = image2.astype('float32')
image4 = image3/255
imageRead.append(image4)
imageArray = np.asarray(image$_read$)

- imread() method loads an image from the specified file. If the image cannot be read (because of missing file, improper permissions, unsupported or invalid format) then this method returns an empty matrix.
- To resize an image in Python, you can use cv2.resize() function of OpenCV library cv2. Resizing, by default, does only change the width and height of the image. The aspect ratio can be preserved or not, based on the requirement. Aspect Ratio can be preserved by calculating width or height for given target height or width respectively.
- astype() method is used to cast a pandas object to a specified dtype, this function also provides the capability to convert any suitable existing column to categorical type. This function comes very handy when we want to case a particular column data type to another data type. Not only that but we can also use a Python dictionary input to change more than one column type at once. The key label in dictionary is corresponding to the column name and the values label in the dictionary is corresponding to the new data types we want the columns to be of.
- normalization is a process that changes the range of pixel intensity values. Applications include photographs with poor contrast due to glare, for example. Normalization is sometimes called contrast stretching or histogram stretching. In more general fields of data processing, such as digital signal processing, it is referred to as dynamic range expansion. The purpose of dynamic range expansion in the various applications is usually to bring the image, or other type of signal, into a range that is more familiar or normal to the senses, hence the term normalization. Often, the motivation is to achieve consistency in dynamic

range for a set of data, signals, or images to avoid mental distraction or fatigue. For example, if the intensity range of the image is 50 to 180 and the desired range is 0 to 255 the process entails subtracting 50 from each of pixel intensity, making the range 0 to 130. Then each pixel intensity is multiplied by 255/130, making the range 0 to 255.

5.3 BUILDING THE MODEL

The model that we propose herein is the sequential model using densenet model as it has proved to give a better accuracy as compared to the others that we tried, i.e., Convolutional neural network model and the VGG16 model. In addition to the densenet model, the global average pooling 2d and a dense layer was also added to the sequential model.

The architecture of the model that we propose herein, is presented below in figure 4.1:

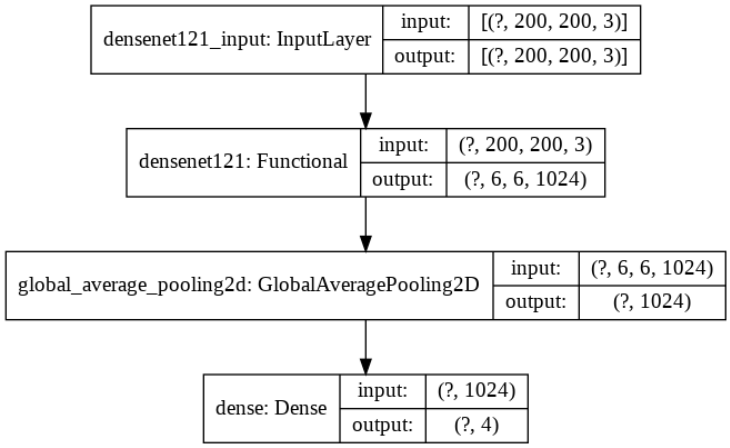

Figure 5.1: Model using densenet

DENSENET121 LAYER:

DenseNet is one of the latest neural networks for visual object recognition. It's quite similar to ResNet but has some fundamental differences.

tf.keras.applications.DenseNet121(
includetop=True,
weights="imagenet",
inputtensor=None,
inputshape=None,
pooling=None,
classes=1000,
)

Arguments:

- includetop: whether to include the fully-connected layer at the top of the network.

- weights: one of None (random initialization), 'imagenet' (pre-training on ImageNet), or the path to the weights file to be loaded.
- inputtensor: optional Keras tensor (i.e. output of layers.Input()) to use as image input for the model.
- inputshape: optional shape tuple, only to be specified if includetop is False (otherwise the input shape has to be (224, 224, 3) (with 'channelslast' data format) or (3, 224, 224) (with 'channelsfirst' data format). It should have exactly 3 inputs channels, and width and height should be no smaller than 32. E.g. (200, 200, 3) would be one valid value.
- pooling: Optional pooling mode for feature extraction when includetop is False.
- None means that the output of the model will be the 4D tensor output of the last convolutional block.
- avg means that global average pooling will be applied to the output of the last convolutional block, and thus the output of the model will be a 2D tensor.
- max means that global max pooling will be applied.
- classes: optional number of classes to classify images into, only to be specified if includetop is True, and if no weights argument is specified.

Returns:

A Keras model instance.

GLOBAL AVERAGE POOLING 2D LAYER:

Now, the outputs of the densenet121 layer was passed as an input to another global average pooling layer which was two dimensional, due to the dataset being of image type.

This block performs exactly the same operation as the 2D Average pooling block except that the pool size (i.e., Horizontal pooling factor x Vertical pooling factor) is the size of the entire input of the block, i.e., it computes a single average value for each of the input channels.

model.add(GlobalAveragePooling2D())

The above code was used to add the layer to add to the sequential model.

Input:

The 2D Global average pooling block takes a tensor of size (input width) x (input height) x (input channels) and computes the average value of all values across the entire (input width) x (input height) matrix for each of the (input channels).

Output:

The output is thus a 1-dimensional tensor of size (input channels).

DENSE LAYER:

Dense implements the operation: output = activation(dot(input, kernel) + bias) where activation is the element-wise activation function passed as the activation argument, kernel is a weights

matrix created by the layer, and bias is a bias vector created by the layer (only applicable if usebias is True).

Note: If the input to the layer has a rank greater than 2, then Dense computes the dot product between the inputs and the kernel along the last axis of the inputs and axis 1 of the kernel (using tf.tensordot). For example, if input has dimensions (batchsize, d0, d1), then we create a kernel with shape (d1, units), and the kernel operates along axis 2 of the input, on every sub-tensor of shape (1, 1, d1) (there are batchsize * d0 such sub-tensors). The output in this case will have shape (batchsize, d0, units).

Besides, layer attributes cannot be modified after the layer has been called once (except the trainable attribute).

model.add(Dense(4, activation='sigmoid'))

The above code was used for adding the dense layer to the sequential model.

Arguments:

- units: Positive integer, dimensionality of the output space.
- activation: Activation function to use. If you don't specify anything, no activation is applied (ie. "linear" activation: a(x) = x).
- usebias: Boolean, whether the layer uses a bias vector.
- kernelinitializer: Initializer for the kernel weights matrix.
- biasinitializer: Initializer for the bias vector.
- kernelregularizer: Regularizer function applied to the kernel weights matrix.
- biasregularizer: Regularizer function applied to the bias vector.
- activityregularizer: Regularizer function applied to the output of the layer (its "activation").
- kernelconstraint: Constraint function applied to the kernel weights matrix.
- biasconstraint: Constraint function applied to the bias vector.

Input shape:

N-D tensor with shape: (batchsize, ..., inputdim). The most common situation would be a 2D input with shape (batchsize, inputdim).

Output shape:

N-D tensor with shape: (batchsize, ..., units). For instance, for a 2D input with shape (batchsize, inputdim), the output would have shape (batchsize, units).

5.4 SPLITTING THE DATASET AND TRAINING THE MODEL

Training data is the observations in the training set form the experience that the algorithm uses to learn. In supervised learning problems, each observation consists of an observed output variable and one or more observed input variables.

The test set is a set of observations used to evaluate the performance of the model using some performance metric. It is important that no observations from the training set are included in the test set. If the test set does contain examples from the training set, it will be difficult to assess whether the algorithm has learned to generalize from the training set or has simply memorized it.

The dataset that we were given was divided into two parts initially, i.e., the train file and the test file. However, the test file did not contain the target labels of the leaves. Henceforth, prediction could not be carried out using that.

Therefore, we had to divide the train file itself into two parts, i.e., the training dataset and the testing dataset. This was done in the ratio of 7:3. The training dataset was 70 percent and testing dataset was 30 percent of the original dataset.

Following was the code, that we used for the splitting purposes using the model selection package and the traintestsplit function from sklearn library. We also fixed the randomstate to be 40.

```
from sklearn.modelselection import traintestsplit
Ptrain, Ptest, Qtrain, Qtest = traintestsplit(imagearray, df, testsize=0.30, randomstate=40)
```

Thereafter, we trained the model built passing the training images and the target labels of the corrsponding images as parameters and for this fit() was used and the epochs were kept to be 100.

```
model.fit(Ptrain, Qtrain, epochs=100, verbose=1)
```

5.5 PREDICTION

Now, after the model is trained on the training dataset, there is a need for the prediction to be carried out, in order to know the accuracy of the model. This is one of the most essential steps when one thinks of the machine learning process.

The predict function is used herein to predict the target labels for the images in the test file. Thereafter, the values obtained after the prediction are stored in a variable. Now, these predicted values are matched with the original target labels of the corresponding images and a classification report is generated.

Many metrics can be used to measure whether or not a program is learning to perform its task more effectively. For supervised learning problems, many performance metrics measure the number of prediction errors.

There are two fundamental causes of prediction error for a model -bias and variance. Assume that you have many training sets that are all unique, but equally representative of the population. A model with a high bias will produce similar errors for an input regardless of the training set

it was trained with; the model biases its own assumptions about the real relationship over the relationship demonstrated in the training data. A model with high variance, conversely, will produce different errors for an input depending on the training set that it was trained with. A model with high bias is inflexible, but a model with high variance may be so flexible that it models the noise in the training set. That is, a model with high variance over-fits the training data, while a model with high bias under-fits the training data.

Ideally, a model will have both low bias and variance, but efforts to decrease one will frequently increase the other. This is known as the bias-variance trade-off. We may have to consider the bias-variance trade-offs of several models introduced in this tutorial. Unsupervised learning problems do not have an error signal to measure; instead, performance metrics for unsupervised learning problems measure some attributes of the structure discovered in the data. Most performance measures can only be worked out for a specific type of task.

Machine learning systems should be evaluated using performance measures that represent the costs of making errors in the real world.

Herein, we use metrics module from the sklearn library in python. This module consists of a function called classificationreport() and this function simplifies our work and generates a detailed report of prediction score and accuracy.

The syntax of the function is as follows:

sklearn.metrics.classificationreport(ytrue, ypred, *, labels=None, targetnames=None, sampleweight=None, digits=2, outputdict=False, zerodivision='warn')

Now, this generates a classification table which gives precision, recall, f1-score and support for all the different classes of data owing to the fact that our data is of multi-class classification problem.

Accuracy, or the fraction of instances that were classified correctly, is an obvious measure of the program's performance. It is the most intuitive performance measure and it is simply a ratio of correctly predicted observation to the total observations. Precision is the ratio of correctly predicted positive observations to the total predicted positive observations. Recall is the ratio of correctly predicted positive observations to the all observations in actual class. F1 Score is the weighted average of Precision and Recall. Therefore, this score takes both false positives and false negatives into account. Intuitively it is not as easy to understand as accuracy, but F1 is usually more useful than accuracy, especially if you have an uneven class distribution. Accuracy works best if false positives and false negatives have similar cost. If the cost of false positives and false negatives are very different, it's better to look at both Precision and Recall.

5.6 TUNING THE PARAMETERS

Every model has a set of keys called parameters which run them. Each parameter is set to a default value which can be changed as per requirement (say, based on some scoring metric).

Now, every real world data set is different and needs to be worked on differently. If the same model with a strict parameter set is applied on every data, a good result cannot be expected uniformly in all cases. Thus, these parameters must be adjusted in such a manner that the best predictions can be achieved by the model for every individual data set which comes by.

This very technique of adjusting the elements which control the behavior of a given model is called parameter tuning.This is the step that is highly responsible for achieving a good accuracy, f1-score, precision and recall. This step is also similar to the hit and trial method as, there is no predefined rule that what parameter value would work for a particular data, and so one needs to constantly try using different values.

Here, in the model, that we propose for foliar disease prediction in the apple leaves,i.e., the densenet model, we changed the activation functions for the dense layer built with four neurons as the data had four target class labels. Firstly, we kept the activation function as 'softmax' and then 'sigmoid'. However, the model gave best accuracy for 'sigmoid'.The pretrained 'imagenet' weights were used as they are highly efficient in grasping the features of the images.

Then, we changed the optimizer from 'adam' to 'rmsprop' as the latter achieves higher accuracy in the classification problems. The loss was kept 'categorical crossentropy' and metrics was defined as 'categorical accuracy'. These were responsible for improving the training accuracy of the model, thereby, making the model more accurate.

Thereafter, the model was trained for a higher number of epochs. All this together, helped us to improve the accuracy of our proposed model.

6. RESULT ANALYSIS

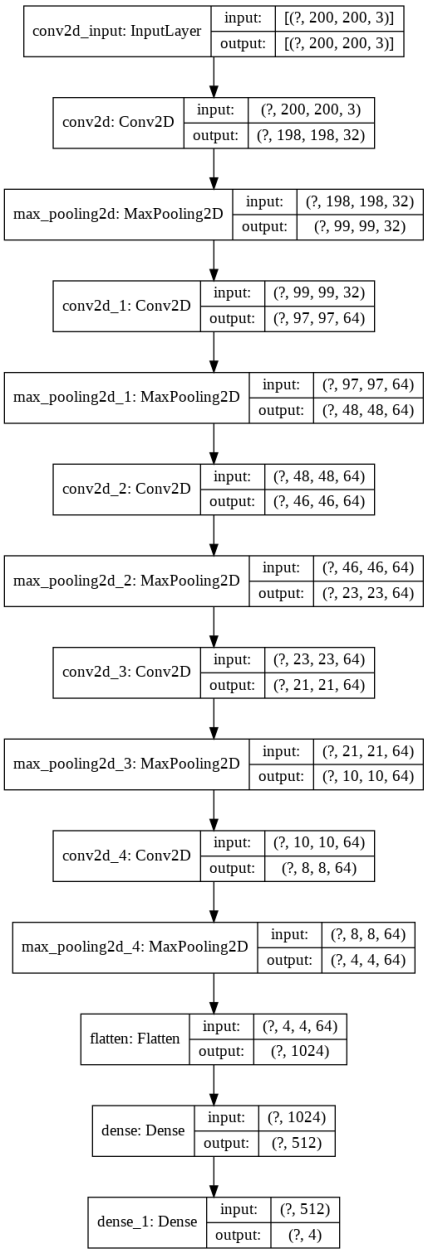

Figure 6.1: Model using CNN

For the dataset, we at first experimented with building a sequential model using convolutional neural network. Here, as presented, in figure 5.1, the model contained several layers in order to achieve better accuracy.

Here, five convolution 2D and five MaxPooling 2D layers were used alternatively, in order to enhance the power of the model. Sixty four Conv-2D of 3X3 stride were used in each layer, and the activation function was relu. 2D was used due to the dataset being of image type.The pool size of MaxPooling2D was (2, 2).

After these layers, a flattening layer was added, thereafter, a dense layer was added of 512 neurons and activation function being relu. At the last, a dense layer was added of 4 neurons, due to the target labels being 4 in number. Here, the activation function used was sigmoid.

After which, it was compiled using Adam optimizer with a learning rate of about 0.001, and was trained on the training dataset for about 200 epochs. After which the prediction was made for the testing dataset and the classification report was therefore generated.

	PRECISION	RECALL	F1 SCORE	SUPPORT
0	0.79	0.14	0.24	134
1	0.00	0.00	0.00	23
2	0.96	0.66	0.78	148
3	0.80	0.19	0.30	151
micro avg	0.90	0.32	0.47	456
macro avg	0.64	0.25	0.33	456
weighted avg	0.81	0.32	0.42	456
samples avg	0.32	0.32	0.32	456

The above table clearly determines the classification report generated by the sequential model with CNN layers.

Later, we tried with another sequential model which consisted of the vgg16 function of keras module. The input shape of the images needed to be changed for the model and it was kept (224, 224, 3). The outputs of the vgg16 layer were sent to the dense layer consisting of 4 neurons.

The architecture of this model is presented in figure 5.2.

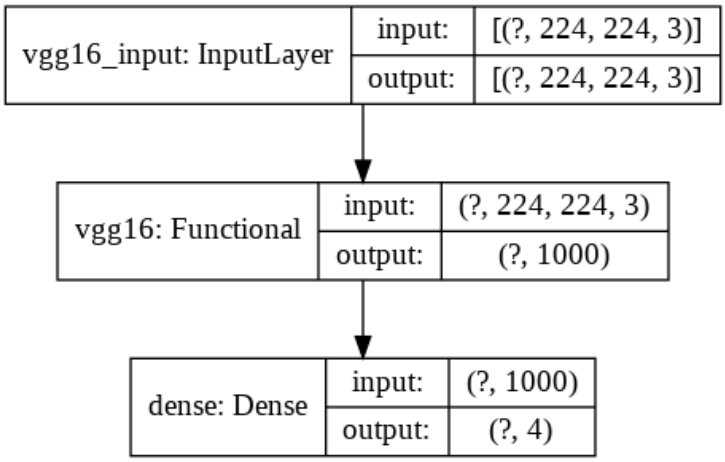

Figure 6.2: Model using VGG16 layer

The model was compiled using Adam optimizer with a learning rate of about 0.001, and was trained on the training dataset for about 200 epochs. Hence, later a classification report was generated as shown below.

	PRECISION	RECALL	F1 SCORE	SUPPORT
0	0.29	1.00	0.45	134
1	0.00	0.00	0.00	23
2	0.32	1.00	0.49	148
3	0.33	1.00	0.50	151
micro avg	0.32	0.95	0.47	456
macro avg	0.24	0.75	0.36	456
weighted avg	0.30	0.95	0.46	456
samples avg	0.32	0.95	0.47	456

Now, since, the accuracy achieved by the above models, therefore, we could not rely on those models and we tried using a different sequential model. This time, we used the densenet121 function as a layer of the model. The output obtained from this layer was feeded as input to the next pooling layer, i.e., the global average pooling 2D.

The outputs from this pooling layer was inputted to the dense layer having 4 neurons, again due to the dataset having 4 target labels.

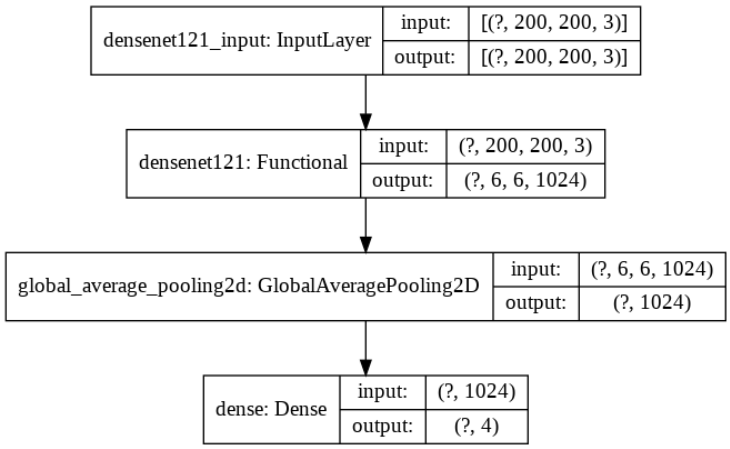

Figure 6.3: Model using densenet121 layer

Here, the densenet121 function used pretrained 'imagenet' weights, which are highly efficient in grasping the eminent features from the images. These had to be downloaded from the googleapis and were 29089792 in total. Its parameter includetop was kept false. The activation function used in the dense layer was sigmoid.

The classification report generated for the model is given below:

	PRECISION	RECALL	F1 SCORE	SUPPORT
0	0.89	0.72	0.80	134
1	0.00	0.00	0.00	23
2	0.99	0.85	0.92	148
3	0.96	0.74	0.84	151
micro avg	0.95	0.73	0.83	456
macro avg	0.71	0.58	0.64	456
weighted avg	0.90	0.73	0.81	456
samples avg	0.73	0.73	0.73	456

Now, after observing the classification report of all the three different types of sequential model, the best accuracy was achieved using the model with densenet121 layer. Therefore, the basic aim of this report is to propose that very same model.

7. DISCUSSION

As a start we have added only few parameters to our machine learning model like in CNN we added only one and two layers but in future we will added more layer to tune our model and get very good accuracy and we will add more features or parameters to our machine learning model which will help the farmers to detect the problems even more faster as well as more accurate. This will change the way of farmers. Currently the dataset used is very limited in dynamics like one thousand eight hundred but in future we will upgrade it and we will work on more amount of dataset. There are numerous of crops and plants of dataset and now we are working on a selected plant but our future plans is include adding specimens of leaf's of various plants and crops. Future dataset be cereals, pulses, rice and various fruits. This will lead to more accurate model in future.

The machine learning model is very useful for farmers to identify the disease very easily because farmers faces very difficulties to detect the disease with their naked eyes. Our machine learning model will help finding out the disease in the initial period of the defected plant, not only for the initial period but it will also help the farmers to detect the upcoming problems for the crop and plant. In now a days various crops are get destroyed by some unknown disease which they can't figure out what type of disease is it but with the help of machine learning model it will be very easy for them to figure out the problem or the disease in time for their crop. Thousand of crop get destroy in a small period of time because of plant disease but our machine learning model will reduce the problems of the farmers, it will ease the life of farmers and will benefit them.

8. CONCLUSION

The utilization of machine learning concepts to plant diagnostics problems is quickly evolving. Although the results obtained in this model are decent, it needs additional work to enhance the model for a feasible use. The dataset was limited to apple plants only, the model needs to be trained with an expanded dataset. Moreover, many problems faced while making this model might be resolved by the expansion of the dataset. For now, the model is trained with leaves image, using other parts of the plant demand some testing. It is also worth mentioning that there are still common hurdles to overcome. The model is a foundation work for smart agriculture, in the future, it will be converted into an API (Application programming interface) and used with a simple web user interface.

References

[1] Morton J Canty. *Image analysis, classification and change detection in remote sensing: with algorithms for ENVI/IDL and Python*. Crc Press, 2014.

[2] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016.

[3] Lars Kai Hansen and Peter Salamon. Neural network ensembles. *IEEE transactions on pattern analysis and machine intelligence*, 12(10):993–1001, 1990.

[4] Wen-Xiong Kang, Qing-Qiang Yang, and Run-Peng Liang. The comparative research on image segmentation algorithms. In *2009 First International Workshop on Education Technology and Computer Science*, volume 2, pages 703–707. IEEE, 2009.

[5] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[6] Donald Michie, David J Spiegelhalter, CC Taylor, et al. Machine learning. *Neural and Statistical Classification*, 13(1994):1–298, 1994.

[7] Mukesh C Motwani, Mukesh C Gadiya, Rakhi C Motwani, and Frederick C Harris. Survey of image denoising techniques. In *Proceedings of GSPX*, pages 27–30, 2004.

[8] Frédéric Patin. An introduction to digital image processing. *online]: http://www. programmersheaven. com/articles/patin/ImageProc. pdf*, 2003.

[9] Robert J Schalkoff. *Digital image processing and computer vision*, volume 286. Wiley New York, 1989.

[10] GT Shrivakshan and Chandramouli Chandrasekar. A comparison of various edge detection techniques used in image processing. *International Journal of Computer Science Issues (IJCSI)*, 9(5):269, 2012.

[11] Patrice Y Simard, David Steinkraus, John C Platt, et al. Best practices for convolutional neural networks applied to visual document analysis. In *Icdar*, volume 3, 2003.

[12] Xin Yang and Tingwei Guo. Machine learning in plant disease research. *European Journal of BioMedical Research*, 3:6, 03 2017.

[13] Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. Recurrent neural network regularization. *arXiv preprint arXiv:1409.2329*, 2014.