T.C.

ISTANBUL AYDIN UNIVERSITY

INSTITUTE OF GRADUATE STUDIES

SMART DETECTION AND DIAGNOSIS OF PLANT DISEASE USING DEEP AND MACHINE LEARNING METHODS

MASTER'S THESIS

Muhammad Umar MURAD

Department of Software Engineering

Artificial Intelligence and Data Science Program

MARCH, 2023

T.C.

ISTANBUL AYDIN UNIVERSITY

INSTITUTE OF GRADUATE STUDIES

SMART DETECTION AND DIAGNOSIS OF PLANT DISEASE USING DEEP AND MACHINE LEARNING METHODS

MASTER’S THESIS

Muhammad Umar MURAD

(Y2013.140004)

Department of Software Engineering

Artificial Intelligence and Data Science Program

Thesis Advisor: Prof. Dr. Ali OKATAN

MARCH, 2023

ONAY FORMU

DECLARATION

I hereby declare with respect that the study “Smart Detection and Diagnosis of Plant Disease Using Deep and Machine Learning Methods”, which I submitted as a Master thesis, is written without any assistance in violation of scientific ethics and traditions in all the processes from the project phase to the conclusion of the thesis and that the works I have benefited are from those shown in the references. (06/03/2023)

Muhammad Umar MURAD

FOREWORD

I would like to express my gratitude to Prof. Dr. Ali Okatan, who served as my adviser for my dissertation, for providing me with all of the necessary advice. I am grateful to him for all of the assistance he has provided during this time. The contributions of his thoughts and support were unquestionably essential and he made it possible for me to improve my thesis and complete it on time. He has always done his best to guide me in the right direction whenever he felt I was stuck.

In addition, I want to take this opportunity to thank my family and friends for providing me with unwavering support and inspiration during the process of completing my thesis. This success would not have been attainable without the involvement of the individuals I have mentioned. Thank you.

March, 2023 Muhammad Umar MURAD

SMART DETECTION AND DIAGNOSIS OF PLANT DISEASE USING DEEP AND MACHINE LEARNING

ABSTRACT

Many tasks are difficult for us but easier for machine to do as to detect any disease in plants it is difficult for human beings to find the diseases in plants without heaving years of experience in farming which can cause immense effect to the plants. In agriculture it is very important to recognize and find the disease of the plants in the early stages. As, disease in the plants can affect the yield of the crops. It is unhealthy for the plants and in return it can affect a lot to the farmer and in last danger to the food-security. Using computer vision techniques, we can classify the plants with the help of state-of the art ML algorithms and deep learning models to differentiate between healthy and the effected plants by classifying their leaves. It is one of the techniques that different researchers worked on different plants using different techniques and different pre-trained deep learning networks (DenseNet121, EfficientNetB0, InceptionV3, VGG19, and Xception) and classic machine learning algorithms to detect the diseases on the plants using the plant leaves.

This research worked on two different approaches. First, the experimental component trains models using different plants images. Then, the training of the models that have already been trained. According to the results that we get in our experiments in our work, the Xception networks performed admirably from deep learning models. It gives us an accuracy of 89.93%. On other hand, from machine learning algorithms Random Forest performed much better than the SVM and Decision Tree. In addition to our results, it indicates that the performance of the pre-trained network system gives the best findings for Plant disease detection.

Keywords: Plant disease detection, plants disease, pre-trained networks, machine learning, deep learning

DERİN VE MAKİNE ÖĞRENMYİ KULLANARAK BİTKİ HASTALIĞININ AKILLI TESPİTİ VE TEŞHİSİ

ÖZET

Bitkilerdeki herhangi bir hastalığı tespit etmek gibi bizim için zor ancak makineler için daha kolay olan birçok görev, insanların tarım alanında uzun yıllara dayanan deneyimleri olmadan bitkilerdeki hastalıkları bulması zordur ve bu da bitkiler üzerinde büyük etkilere neden olabilir. Tarımda, bitkilerin hastalıklarını erken aşamalarda tanımak ve bulmak çok önemlidir. Çünkü bitkilerdeki hastalıklar mahsulün verimini etkileyebilir. Bitkiler için sağlıksızdır ve karşılığında çiftçiyi çok fazla etkileyebilir ve son olarak gıda güvenliğini tehlikeye atabilir. Bilgisayarla görme tekniklerini kullanarak, yapraklarını sınıflandırarak sağlıklı ve etkilenmiş bitkileri ayırt etmek için en son makine öğrenimi algoritmaları ve derin öğrenme modelleri yardımıyla bitkileri sınıflandırabiliriz. Bitki yapraklarını kullanarak bitkilerdeki hastalıkları tespit etmek için farklı araştırmacıların farklı teknikler ve farklı önceden eğitilmiş derin öğrenme ağları (DenseNet121, EfficientNetB0, InceptionV3, VGG19 ve Xception) ve klasik makine öğrenimi algoritmaları kullanarak farklı bitkiler üzerinde çalıştığı tekniklerden biridir.

Bu araştırma iki farklı yaklaşım üzerinde çalışmıştır. İlk olarak, deneysel bileşen farklı bitki görüntülerini kullanarak modelleri eğitmektedir. Ardından, daha önce eğitilmiş olan modellerin eğitilmesi. Çalışmamızda yaptığımız deneylerde elde ettiğimiz sonuçlara göre Xception ağları derin öğrenme modellerinden takdire şayan bir performans sergilemiştir. Bize %89,93'lük bir doğruluk oranı veriyor. Diğer yandan makine öğrenmesi algoritmalarından Random Forest, SVM ve Decision Tree'den çok daha iyi performans gösterdi. Sonuçlarımıza ek olarak, önceden eğitilmiş ağ sisteminin performansının Bitki hastalığı tespiti için en iyi bulguları verdiğini göstermektedir.

Anahtar Kelimeler: Bitki hastalığı tespiti, bitki hastalığı, önceden eğitilmiş ağlar, makine öğrenmesi, derin öğrenme

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

AI	: Artificial Intelligence
AUCROC	: Area Under the Curve of Receiver Operating Characteristic
BN	: Batch Normalization
BPNN	: Back Propagation Neural network
CNN	: Convolutional Neural Network
Conv	: Convolution
ConvNet	: Convolutional Network
CCV	: Color Coherence Vector
CAM	: Class Activation Maps
CART	: Classification and Regression Trees
DL	: Deep Learning
DCNN	: Deep Convolutional Neural Network
GAN	: Generative Adversarial Networks
HSV	: Hue, Saturation, Value
ILSVR	: ImageNet Large Scale Visual Recognition Challenge
IG	: Information gain
ICT	: Information and Communication Technology
IEEE	: Institute of Electrical Engineering and Electronics Engineer
KNN	: K-Nearest Neighbors
ML	: Machine Learning
NAS	: Network Attached Storage
NLP	: Natural Language Processing
RGB	: Red, Green, Blue
PRAUC	: Precision-Recall of Area Under the Curve
RBF	: Radial Basis Function
RDF	: Resource Description Framework
ReLU	: Rectified Linear Unit
STD	: Standard
UV	: Ultraviolet
VGG19	: Visual Geometry Group 19
VGGNet	: Visual Geometry Group Network

SVM	: Support Vector Machine
UNFAO	: United Nations Organization of Food

LIST OF FIGURES

LIST OF TABLES

LIST OF EQUATIONS

I. INTRODUCTION

In contemporary times, agricultural field is vital to the economies of some countries, including Pakistan, India, and Russia (Merlina et al. 2021). Due to the severe economic losses caused by plant diseases, tracing their origins is of paramount importance. Even though symptoms of a disease may show up in other plant parts, leaves are often examined first. Plant diseases account for around 20% of crop losses globally, despite international efforts to increase food security and decrease plant mortality. Pollution and climate change have made this an immediate concern since the turn of the decade. National and global food supply networks substantial economic costs because of production losses due to plant diseases. According to the International Plant Protection Convention (2017) put forth by the UNFAO, plants insects, pests and illnesses causes the loss of 20–40% of world food output (Ouhami et al. 2021). One-thirteenth of annual agricultural losses can be attributed to plant diseases. These results highlight the significance of accurate plant disease diagnosis in preventing unnecessary production losses. A global loss in agricultural productivity of up to 13% is blamed on plant diseases, according to some estimates. These results highlight the importance of accurate plant disease identification in avoiding costly output losses. Locating and analyzing disease origins in plants should be a top goal. We need to start by figuring out what causes plant sickness.

A pathogenic infection, a susceptible host, and favorable environmental conditions are the three essential components for a plant to become unwell. The triangle of plant disease formed by these three elements is seen in Figure 1. Most plant illnesses grow up the plant from the roots. Infected plants can quickly spread the disease to uninfected ones. For the early finding and treatment or taking precautionary measures of crop diseases, ongoing surveillance is crucial. However, many plant diseases appear either at the same time as pollination or after the season's harvest has already taken place. Plant diseases can be further categorized based on the primary organs that they affect. Because the initial signs of foliar diseases manifest on the leaves themselves, plant pathologists rely heavily on visual inspection to make diagnoses. It is estimated that fungal infections cause 30–50% of all crop failures. For this reason, researchers are using different techniques based on Artificial Intelligence build fully automated systems that can diagnose a plant's issues simply by glancing at a photo of its sick leaves (Elhassouny and Smarandache 2019). A few of the many crucial issues related to plant disease diagnostics include early season diagnosis; simultaneous detection in different crops; evaluation of severity; estimation of pesticide treatment volume; and practical tactics in controlling illness to minimize its spread.

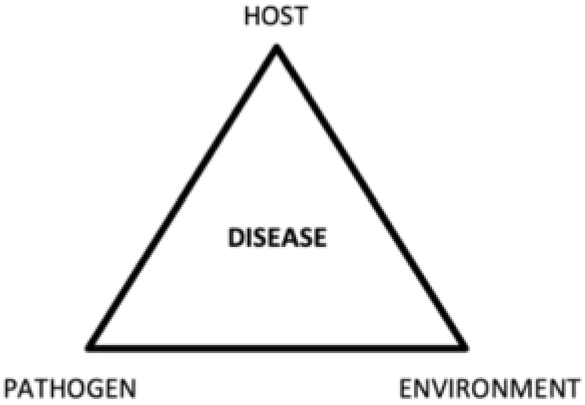


Figure 1 Triangle of Plant Disease (Ahmad, Sarawat and El Gamal 2023).

Accurate and timely disease detection in plants is critical for controlling their destructive potential. An incorrect diagnosis might cause ineffective management choices, such as the application of the wrong chemical, which can further compromise health and reduce yield. Before the advent of modern farming equipment, farmers would send plants to a diagnostic laboratory, where professionals would recommend a treatment. Now, however, farmers are more likely to consult online databases of plant diseases or local pathologists for guidance. Because of the proliferation of mobile phones, many researchers have worked to advance the production in agricultural development productivity by employing ICT resources. However, the conventional method of diagnosis relies solely on one's own eyes are inefficient and particularly when applied to vast areas (Javidan et al. 2023). Having to pay for specialists' constant upkeep is a major drawback. An intelligent disease detection system could be helpful to quickly identify plant illnesses.

Scientists in the agricultural industry have started to investigate vision's possible usefulness in recent years. Weed control, disease detection, robotic harvesting, fruit ripeness assessment, insect identification, and more have all benefited from these developments. Computer vision may make use of a wide variety of cameras, including color, multispectral, and even hyperspectral varieties. Visible-light sensors are typically used in machine vision applications. A plethora of other CNNs have been introduced in studies for plant disease identification (Ouhami et al. 2021). There are several tools and programs available to reduce the severity of plant diseases. Even though some possible ways and tools have been suggested for finding plant diseases, more research needs to be done.

In this research, we analyzed 70 studies to determine how they contributed to our understanding of plant diseases by doing things like developing identification and severity estimation models for plant diseases, where they used sensors and worked on the improvement of the existing architecture or filling in gaps in the literature. Google Scholar, IEEE Xplore, Science Direct, and Scopus were combed for articles on plant diseases detection, for precision agriculture, agriculture drone systems, image data sets, object recognition, semantic segmentation, picture classification, machine learning, deep learning, and transfer learning. This research documents the development of several strategies for identifying diseases, as well as new forms of data, imaging sensors, and techniques for collecting information. We also examine the gaps in the existing literature to assist us in determining how to proceed with creating a method for farmers to control illness in real-time using deep learning. Each part of this study was carefully planned to answer important questions in the field of diagnosing plant diseases.

I was wondering what common open-source datasets are used to investigate plant diseases. What are some tried-and-true methods for taking photos of diseases, such as imaging sensors and data aggregation platforms? How are plant diseases detected using deep learning techniques? How good the deep learning models performed when used with various test data, datasets, and field photography? Is it possible that deep learning will one day be able to diagnose plant diseases just as well as experts?

Using state-of-the-art AI techniques including image processing, ML, and DL, this research proposes a couple of deep learning models and classic machine learning algorithm for accurately diagnosing plant diseases. Later, we compared the effectiveness of the proposed models and algorithms to get a sense of its efficacy.

A. Motivations

There are many factors contribute to research on this work. The capacity of this classification task can be implemented in different ways. Ranging from the real time disease detection to the agricultural drone it has various application and uses. It can also be utilized on the smart phones to detect the real time damage to the plants (M. Bhange et.al, 2015) Main aim is to help the farmers in the certain field to make their work progressive and also, they can help in securing the food shortage and also, they can help the economy of their countries to become strong. This area of farming not only includes a lot of expenditure, but it is also one of the hard-working fields. After a lot of hard work and spending time if farmer is getting losses because of these diseases to their plants and crops, it gives a big damage to the farmers.

Thanks to the advancement of the artificial intelligence that is helping us in a lot of disciplines. Now, there are a lot of state-of-the-art models that can be utilized for the classification tasks that can help the farmers to increase the growth rate of their crops. (David Rodman, 2019) write about the success of the agriculture field. Israel has a very small land as compared to the big nations like Pakistan and India which are basically agriculture countries but with the advancement and use of modern tools especially their agricultural drones where they can check the aerial view of their crops and inspect the plant leaves using good quality of camera and the ai to detect any damage or insects on the plants. Modern techniques are very important. These techniques cannot only help to reduce the human efforts, but it can also help to boost the yield and efficiency.

B. Aims and Objectives

This research employs machine learning and deep learning models that have been pre-trained using images of different diseases of various plants. The primary objectives are described as follows:

- This work Provides various techniques to locate the infected areas of the plants in the early stages to prevent the plant to get more damages
- In this study, we utilize eight different machine and deep learning pre-trained networks to assure smoothness.
- To examine the plant disease detection, we utilize different models that were trained with different plant diseases of various species.

- Images were pre-processed to a specific ratio for deep learning models and machine learning algorithm as models needed a fix-size input
- This work's robustness is evaluated by applying it to different images and to measure its performance we used variety of performance indicators.

C. Thesis Outline

The thesis has the structure shown below:

- Chapter 2 is about the literature review where we studied different authors that worked on the same kind of classification problems and extended our work by studying their models and data sets.
- Chapter 3 is for the models that we give the priority for this thesis according to the research and study. Currently, plant disease classification methods are described in a brief study.
- Chapter 4 Shows conclusion of our results of testing and data analysis. It also defines the accuracy of our task.
- In the Chapter 5 represent our conclusion. Where we have explained our findings and the recommendation for further research.

II. BACKGROUND STUDIES AND RELATED WORKS

Plant disease detection has been researched extensively using a variety of techniques and methodologies, there are more recent of which are described below:

Marko et al. (Darko et al., 2019) used the Data set with around 80 thousand leaves pictures taken in different angles and in different lightning condition so that the real time results can be better for the application. In that paper they used two augmentation strategies. They used augmentation and GAN. With the following they proposed the PlantDiseaseNet to find the disease in a plant. They get an accuracy of 93.67%. Goal of paper (Muammer et al., 2021) was to identify Disease detection using SVM (Support Vector Machine) classifier. They fine-tuned and combined six CNN and then evaluated them on the problem both with ensemble and individually. With ensemble model they obtain an accuracy of 97.56% and they get the accuracy of 96.83% using early fusion ensemble model.

M.Bhange et al. (2015) created a web-based tool which by uploading an image to this tool, it can identify fruit diseases. [5]. Features extraction has been implemented using morphology, color, and CCV characteristics (color coherence vector). The k-means algorithm has been used for clustering. Fruits that are diseased or not are classified using the SVM (Support Vector Machine) classifier. They have an 82% success rate in correctly identifying the disease on pomegranate. In order to distinguish the affected area and classify diseases, H. Ali et al. (2017) applied the color difference algorithm and used textural and color histogram features. Different classifiers, including KNN, Cubic SVM, and Bagged tree classifiers, have been employed in their work. With an accuracy of 99.5%, the bagged tree classifier performs better than the competition.

It was suggested to use Deep Convolutional Neural Network to categorize the four different cucumber plant diseases by (Ma et al, 2018). To increase the dataset and prevent it from overfitting, they used the Data Augmentation approach in their research. They achieve an accuracy of 93.4% when DCNN is considered. Additionally, SVM, Random Forest, and AlexNet comparative tests are carried out by (Ma et al., 2018). Comparative data also demonstrates DCNN's superior performance.

Based on the visual symptoms, S. S. Sannakki et al. (2015) proposed the "Classification of Pomegranate Diseases Based on Back Propagation Neural Network (BPNN)". The Bacterial Blight and Wilt Complex were the primary study topics by removing features, such as color and texture. They were successful in obtaining an accuracy of 97.30%. P.R. Rothe et al. (2015) studied the effects of bacterial blight, myrothecium, and alternaria on cotton plant leaves. Authors employed a technique called pattern recognition. They achieve an accuracy of 86.52% for their classification by employing the Back Propagation Neural Network as their classifier. The effective method to isolate the illness spot is to use the snake segmentation algorithm. However, it also has a drawback due to the slow processing.

Banana Bacterial Wilt and Banana Black Sigatoka, two diseases of the banana plant, were the focus of Godliver O et al (2019) research. To improve the outcomes, multiple extraction methods were applied. To identify the best outcomes, they applied seven different classifiers and extracted various features. It also covers the conversion of the RGB and RGB into HSV and L*a*b, respectively. They employed Naive Bayes, SVM, Random Forest, Nearest tree, Extremely randomized tree, Decision tree, and Nearest tree classifiers. The highly randomized tree performs the best, according to the comparison analysis. The work on an SVM-Based Multiple Classifier System for the detection of Wheat Leaf Diseases is presented by Uan Tian et al. (2012). The features are divided into three groups: color features, texture features, and form features. These traits were divided into three categories: low-level, mid-level, and high-level. The SVM is trained at a high level and provides the best recognition outcome. Sladojevic et al. (2016) employed a deep neural network-based model for classifying leaf images. They created a model that can assess 13 different plant disease kinds from within healthy leaves. They trained the deep convolutional network using a framework created by the Berkley Vision and Center, dubbed "Café." Depending on the different classes, their constructed model was able to provide a precision of between 91 and 98%. The accuracy was 96.3% on average.

In order to explore the analysis of plant pictures based on thermal and visible light, Raza et al. (2015) used an approach. They employed the first stage to register the images in both visible and thermal light. For the registration, they utilized the silhouette extraction. In this research, they put forth an algorithm for registering thermal and visible light images. This registration method demonstrates great accuracy.

Romer et al. (2011) focused on identifying the pathogen infection on the wheat leaf in the initial phases in order to boost output. Pre-symptomatic illness is their main area of interest. The classification of the data was done by the authors using a Support Vector Machine. They employ the laser fluoroscope, where information induced by UV light was gathered. They were able to attain 93% accuracy using the SVM. One advantage is that the outcomes can be seen as early as the second day after the vaccination, even before any physical symptoms manifest. Using deep learning, Chohan et al. (2020) focused on the detection of the plant. They worked with the dataset known as "PlantVillage." They put forth a deep learning-based algorithm that can identify various plant diseases. Data augmentation to expand the sample size was the first stage in developing their model. Then, several convolution and pooling layers were applied to CNN. Their model has a 98.3% accuracy rate.

A public dataset was used by Sharada et al. (2016) to identify 26 illnesses and 14 crop species. They employ DCNN, and the accuracy of their suggested model was 99.35%. They concentrated on AlexNet and GoogleNet, two DCNN architectures created for the "Large Scale Visual Recognition Challenge" (ILSVRC). Using photos of healthy and diseased plants, Ferentinos (2018) focused on the disease identification and diagnosis. The dataset used by the author contained 25 different plants divided into 58 different classifications. The suggested model creates an automated system for the identification and diagnosis of the plant disease by using a particular CNN architecture that has been trained and evaluated. Their highest performance was 99.53%.

The leaf images of tomato plants are used by Chowdhury et al. (2021) to detect tomato infections. Different CNN-based architectures were worked on by the authors. They compare how well the design performs depending on how many classes it has, such as binary classification, six-class classification, and ten-class classification. The majority of architectures produce positive results, but EfficientNet-results B7's stand out because of its accuracy of 99.95% when classifying objects into six categories. With ten classes, EfficientNet-B4 achieved an accuracy of 99.89%. They conclude that architectures can perform significantly better when they are trained on segmented images using deeper networks.

Table 1 A summary of recent plant disease detection research

Study	**Classifier**	**Result/Accuracy**
(Marko et al., 2019)	GAN	93.67%
(Muammer et al., 2021)	SVM	97.56%
(Bhangee et al., 2015)	SVM	82%
(H.Ali et al., 2017)	Bagged tree	99.5%
(MA et al, 2018)	DCNN, Random Forest, SVM	93.4%
(S. S. Sannakki et al,2015)	BPNN	97.30%
(P.R. Rothe et al,2015)	BPNN	86.52%
(Sladojevic et al,2016)	DCNN	96.3%
(Raza et al, 2015)	SVM	90 %
(Romer et al, 2011)	SVM	93%
(Chohan et al, 2020)	CNN	98.3%
(Sharada et al, 2016)	DCNN	99.35%
(Ferentinos, 2018)	CNN	99.53%
(Chowdhury et al., 2021)	CNN	99.95%
Study	Classifier	Result/Accuracy
(Marko et al., 2019)	GAN	93.67%
(Muammer et al., 2021)	SVM	97.56%

It is challenging for humans to diagnose disease in plants. Either they are unable to monitor every plant's leaf or its roots, or it is challenging for them to distinguish.

Farming needed skilled workers. Farming experience is too crucial. In order to get beyond these restrictions and to offer effective strategies to boost productivity while lowering labor costs. Using deep learning networks and algorithms, this study presents a disease categorization approach that illustrates the numerous diseases that can affect diverse plants.

III. APPROACH AND SYSTEM DEVELOPMENT

The step-by description of the procedure of the proposed system is as follows: image acquisition, preprocessing of the input image, model training, and plant disease classification.

In the image acquisition phase, as per research, the other related studies to this topic were using mostly similar datasets, so the author of this study decided to create his own new dataset gathered from the different resources having different 20 categories of healthy and diseased plant leaf images I.e., Cassava Mosaic Disease, Cassava Healthy, Cassava Brown Streak Disease, Cassava Green Mottle, and Cassava Bacterial Blight, Corn Blight, Corn Common Rust, Corn Gray Leaf Spot, Corn Healthy, Cotton Diseased, Potato Early Blight, Potato Healthy, Potato Late Blight, Cotton Healthy, Pepper Bacterial Spot, Pepper Healthy, Rice Brown Spot, Rice Healthy, Rice Hispa, Rice Leaf Blast.

During the pre-processing procedure, 2 methods were used. One method is scaling images to allow models to accurately predict the data and the second method is one hot encoding by which it changes categorical data in order to feed into the machine learning and deep learning algorithms, It increases the precision of prediction and classification. The data, which comprises images of different plant leaves, is then put into a number of classifiers that are based on pre-trained deep learning models (DenseNet121, EfficientNetB0, InceptionV3, VGG19, and Xception) and ML classical algorithms (Decision Tree, Random Forest, and SVM) to assess disease detection based on a plant leaf image.

A. Dataset Analyzation

The plant disease detection models that we utilize were trained with 15852 images of various plant species (see Figure 2) that have been infected with a number of diseases so that we could evaluate how accurate the suggested models are. The collection of the dataset is fairly broad; different categories contain images of various sizes and quantities. The images were pre-processed to a ratio of (256x256) pixels for deep learning pre-trained model and (50x50) pixels for machine learning classical algorithms since our models needed a fixed-size input.

We utilize these images in the train set and test set with the ratios of 70/30 to train our classifiers and correctly assess the accuracy of each model. Table 2 contains other information that pertains to the dataset that was obtained.

Table 2 Details on the dataset, including the breakdown of data into training and testing categories.

Plant Type	**No. of Images**
Cassava bacterial blight	415
Cassava brown streak disease	497
Cassava green mottle	410
Cassava healthy	573
Cassava mosaic disease	365
Corn blight	1145
Corn common rust	1306
Corn gray leaf spot	574
Corn healthy	1162
Cotton diseased	1103
Cotton healthy	848
Pepper bacterial spot	719
Pepper healthy	1065
Potato early blight	1303
Potato healthy	816
Potato late blight	1132
Rice brown spot	377
Rice healthy	1072
Rice hispa	408

Rice leaf blast	562
Total No. of Images:	15852
No. of Images in Training Set:	11097
No. of Images in Test Set:	4755

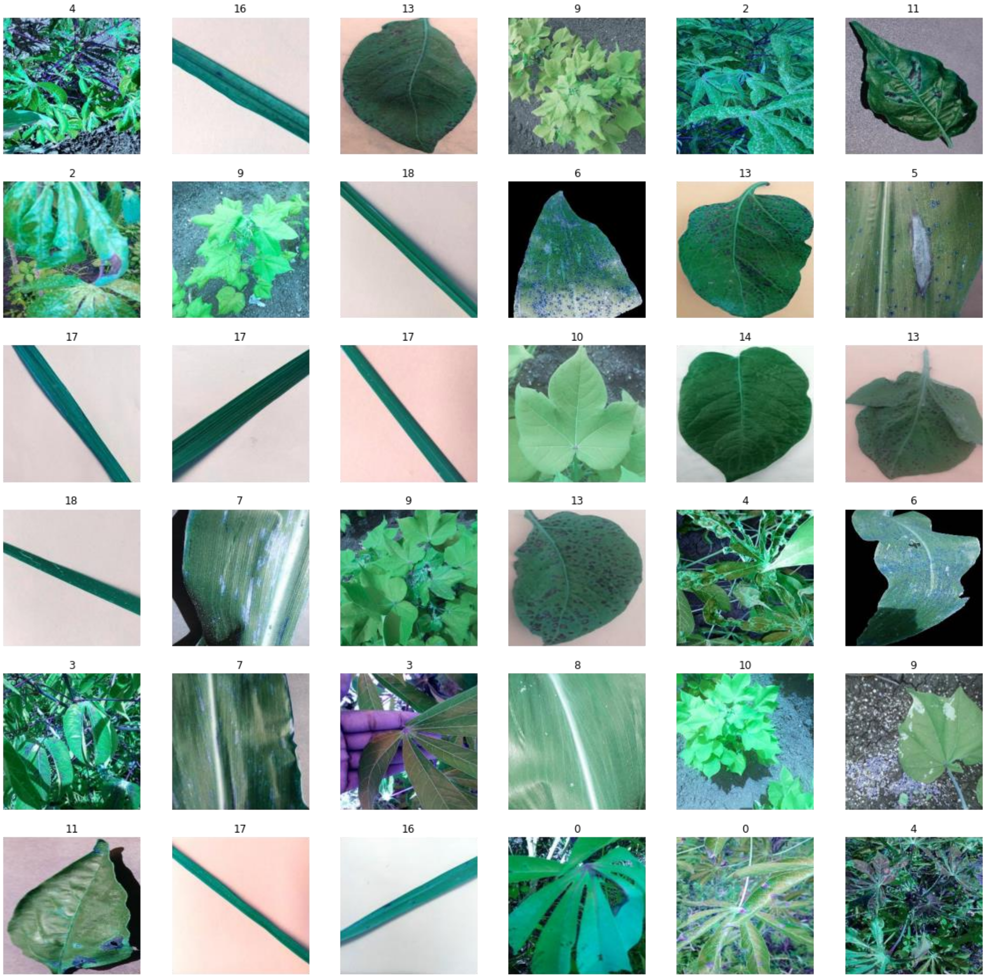


Figure 2 Sample of different healthy and diseased 20 types of plant leaves images

B. Dataset Pre-processing

To generate better results from our models, we must first prepare the data. Only then will we be able to correctly input the data into our models. As a result, we do two phases of data preparation on the dataset. The first stage is scaling the images to the suitable fixed ratio and shrinking the value distribution for the suggested models and algorithms. In the second phase, we change our data (categorical variables) into a format that deep and machine learning algorithms could use to make better predictions. The next subsections will offer a brief summary of the details of each indicated step that we will be employing for this work.

1. Scaling

All the images from the dataset used in this work will need to have their dimensions adjusted to a predetermined size before being fed into our pre-trained models and algorithms. This is because used neural networks and algorithms prefer their inputs to be of the same size. Therefore, in this first phase of the preprocessing method, we are resizing the images to a fixed resolution of 256 by 256 in order to feed them into the proposed pre-trained models and a fixed resolution of 50 by 50 for the machine learning algorithms. This is done in order to fulfill the prerequisites that are specified by the proposed models.

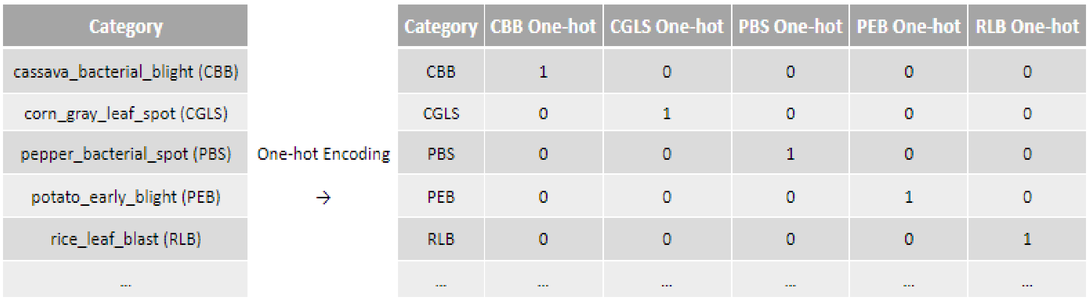
Furthermore, in order to standardize the functional range of the input dataset and efficiently predict the disease category in plant leaf images using random forest, decision tree, and SVM models we first scaled and reorganized our data before feeding it into the mentioned algorithms. The Standard Scaler was then used to resize the value distribution to the point where mean value for the observed data is 0 and variance is 1. Standard incision of a sample "s" is calculated as shown in Equation 1 where "m" shows training samples' if with mean attribute is equal to false, and the training samples' standard deviation is "d," which is one if the with std attribute is false.

$$x = \frac{s-m}{d} \qquad (1)$$

2. One-hot Encoding

As One-hot encoding is an important part of feature extraction for machine learning, to effectively prepare our data for the proposed algorithms and models to get a better prediction, we use a one-hot encoding method to convert each of our categorical values to a new column of categories. Also, it allocates binary value either as 1 or 0 to all those columns. Using this method each of our integer values is depicted a binary vector. The index is indicated by 1 and all the values are zero.

In this work, the categorical data depicts the values which comprises of label values. For example, each of the "plant leaf disease category" variables could have the values from 20 different categories; "cassava_bacterial_blight", “cassava_brown_streak_disease”, “cassava_green_mottle”, “cassava_healthy”, “cassava_mosaic_disease”, “corn_blight”, “corn_common_rust”, “corn_gray_leaf_spot”, “corn_healthy”, “cotton_diseased”, “cotton_healthy”, “pepper_bacterial_spot”, “pepper_healthy”, “potato_early_blight”, “potato_healthy”, “potato_late_blight”, “rice_brown_spot”, “rice_healthy”, “rice_hispa”, “rice_leaf_blast” see Figure 2 that depicts the mapping of one-hot encoding and assigning 0 or 1 to some of the plant leaf disease categories.



Category
cassava_bacterial_blight (CBB)
corn_gray_leaf_spot (CGLS)
pepper_bacterial_spot (PBS)
potato_early_blight (PEB)
rice_leaf_blast (RLB)
...

One-hot Encoding

→

Category	CBB One-hot	CGLS One-hot	PBS One-hot	PEB One-hot	RLB One-hot
CBB	1	0	0	0	0
CGLS	0	1	0	0	0
PBS	0	0	1	0	0
PEB	0	0	0	1	0
RLB	0	0	0	0	1
...	...	...	...	...	...

Figure 3 Sample of one-hot encoding method implemented in the proposed dataset

We generate a binary vector that reflects our numerical values after assigning numeric values. In this case, our vector will have 20 as its length since we have 20 values (categories). Thus, in the example shown in Figure 2 the CBB values are shown by binary vector [1,0,0,0,0], CGLS as [0,1,0,0,0], PBS as [0,0,1,0,0], PEB as [0,0,0,1,0], RLB as [0,0,0,0,1] and so on.

C. Machine and Deep Learning Models

1. Decision Tree

There is different type of classifications that are used to classify the data by categorizing it in different categories or by the addition of the label. Decision tree is one of the classifications where it classifies the data starting from the root node and reaches at the end node which is also called as the leaf node. This is the reason that it is called as the decision tree. This tree keeps on growing with the possible number of solutions or conditions coming. It can be used for classification and also it can also be used for the regression (Chauhan, 2022). For decision tree, we need the data. As data is required to train the machine by the supervised learning. Decision tree works in a very simple way. If there is a certain task to do, we can create a decision tree. What it will do, it will make some certain attributes from the data sample, and it will gives us the most useful attribute. For example, the task ‘Hungry man’ or the root of the tree. Now, this root will grow some branches as a solution for this root. Either this man wants to eat fish or chicken. It will lead to another branch.

It will check that rather he has the money to buy chicken or fish. Then, the person will choose that with which option I need to go. So, decision tree also finds the maximum possible solution for a solution and decide the best for a certain problem. The process is recursive and iterative. It works until get the maximum. When it is stuck into a certain point where two attributes come with same properties and a question arise that where to move next, it uses an algorithm called CART. This algorithm uses the metric of Gini-impurity at single node. There are different terminologies that are used in the decision tree. Root nodes which basically shows the sample of the dataset. Leaf node, which is the resultant of the possible solution, and it cannot be further divided into the node. Splitting is also used which means to divide into sub nodes as per the conditions.

Table 3 Table of different attributes

Day	**Weather**	**Temperature**	**Humidity**	**Wind**	**Play?**
1	Sunny	Hot	High	Weak	No
2	Cloudy	Hot	High	Weak	Yes
3	Sunny	Mild	Normal	Strong	Yes
4	Cloudy	Mild	High	Strong	Yes
5	Rainy	Mild	High	Strong	No
6	Rainy	Cool	Normal	Strong	No
7	Rainy	Mild	High	Weak	Yes
8	Sunny	Hot	High	Strong	No
9	Cloudy	Hot	Normal	Weak	Yes
10	Rainy	Mild	High	strong	No

a. Entropy

It is a metric that is used to find the purity or impurity of a dataset. The purity and impurity lead us to the correct node to proceed with our process of reaching the final node. (Saini, 2021). This metric is used to check especially the impurity in a node.

Here, important thing to understand the is the impurity. Impurity predicts that how much randomness we have in our data. For example. The following graph shows the value maximum as 1 and minimum as 0. So, when the data is highly random or impure then the value 0.5 or 1. But, when the data is pure and there is no randomness in the data then the value is equal to zero. p+ positive class (Probability). p– negative class (Probability). S Sample space. This equation (2) is used to find the entropy.

$$E(S) = -P_{(+)}\, logP_{(+)} - P_{(-)} logP_{(-)} \qquad (2)$$

Mathematically, if we find the impurity of the graph at value 0.5 from the above-mentioned equation, it will give us a value equal to zero. It shows that at 0.5 it has the maximum impurity for the data sample.

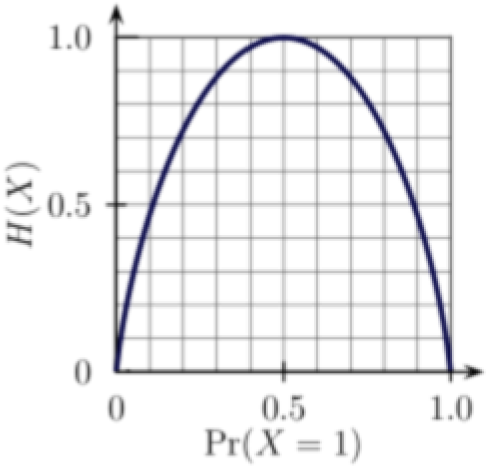


Figure 4 Entropy Range (Brona,2007)

Entropy means that if we are heaving high value for it, it means that the more data is impure. When it has low value so the data is less impure.

$$E(s) = \sum_{i=1}^{c} -Pilog2Pi \qquad (3)$$

If there are more than one (multiple entropies), then the entropy for these attributes can be represent like this as shown in (4).

$$E(T, X) = \sum_{c \in x} P(c)E(c) \qquad (4)$$

b. Information Gain (IG)

To build a decision tree means to find the best attributes which can yield maximum information gain. When we have different attributes then the IG decides that which decision node should be chosen. It is gained according to the reduced impurity in a particular node. Information gain depicts the correct value after splitting the root node and it calculates the value as an average of the entropy before and after the splitting of the dataset. It is used in the decision tree during the iterations.

$$Information\ Gain = i - i_{children} \qquad (5)$$

In decision tree, there are different terminologies. There is a node from which another node gives a birth, and this node is called the parent. The one that was originated or comes from the parent becomes the child node. Along that there are some different terminologies used as sub-tree and pruning. Information gain is used when we need to find that how good these nodes are divided in a decision tree. Information gain is symbolized by the phrase Gain. There the entropy for the parents shown by i and entropy for children as I children. To understand information gain and its computation, The parent (root) node should be chosen based on the property from the set that provides the most information gain. It is attribute A, as seen in figure (5).

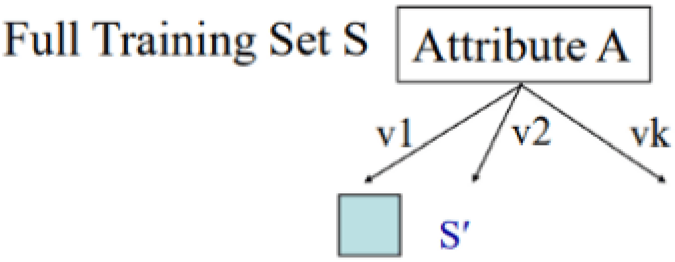


Figure 5 Information gain to build decision tree (Collins A)

c. Gini Impurity

In order to identify how the characteristics of a data collection should divide nodes to create the tree, decision trees are constructed using the Gini Impurity measurement (Chauhan, 2022). In more detail, the Gini Impurity of a data set is a value between 0-0.5 that represents the probability that fresh, random data will be incorrectly categorized if a random class label is assigned to it in accordance with the data collection's class distribution. It is used in Cart algorithm in decision tree to find the impurity index or Gini index and it is a cost function. This is used to split the data sets.

$$Gini = 1 - \sum_{i=1}^{c} (P_i)^2 \qquad (6)$$

d. Punning decision tree

To the extent that the halting requirements are not met, the splitting process produces fully formed trees. However, a fully formed tree is probably going to overfit the data, which will result in low accuracy on unobserved data. Pruning do a job to remove the unwanted data or the unwanted branches from a tree. Pruning is useful so that it can removes the unwanted attributes from the data and we can get to the right attribute and reach our leaf node. Figure 4, shows that how it reduces the unwanted attributes.

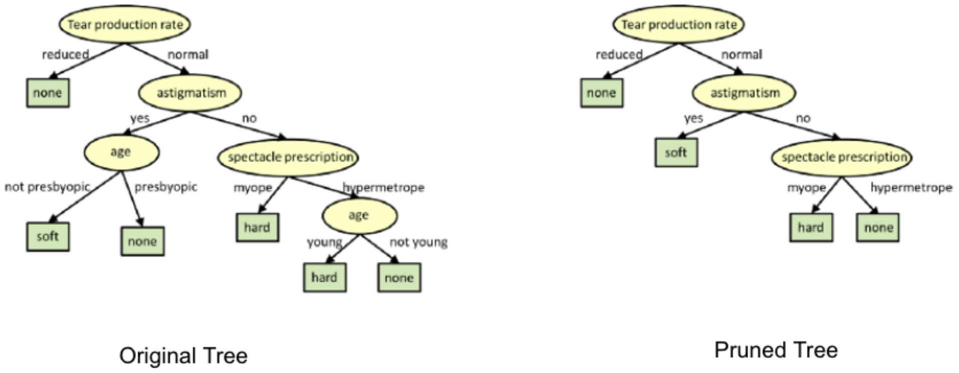


Figure 6 Pruning (Bhiksha)

2. DenseNet

Each of the layers are linked to each other layer in a DenseNet design, as they are densely connected so they are called as the dense network. (Gao Huang, 2017). It comes in several variants. This algorithm, which is used for classification, was created to address the Vanishing gradient problem and boost model accuracy. A typical issue with deep neural networks is the vanishing gradient problem. When there are numerous hidden layers in a deep neural network between the input and output layers. Because of the great distance between the input and output, the majority of the information is lost before it reaches the output. By directly connecting one layer to the next, DenseNet overcomes the problem of the vanishing gradient and provides the best accuracy.

a. Architecture

It comprises of different blocks and transition layers. In DenseNet the input goes directly into the convolutional network and there are uses a filter which have a size of 7*7 and with stride of size 2. Later, pooling layer come which have a size of 3*3 and stride which have a size of 2. DenseNet made up of four different blocks. Each block has different number of convolutional layers. Block one contains six convolution layers. Block two with 12, three with 24 and fourth block have 16 layers. Among each of the dense block it contains a transition layer. Inside each convolutional layer it contains Batch norm, Relu, conv 3*3 and the drop out. While on the other hand the transition layer contains the Batch norm, Relu, conv 1*1, drop out and the pool of size 2*2.

Features maps becomes the input for next layer (Gao Huang, 2017). Inside each dense block, every convolution layer is connected to other layer directly and also share the feature map with each other. One layer provides the feature map of it to the other layer and the other layer put some feature on the top of it. Inputs features are concatenated with each layer. For this the size of each layer must be equal. As, the convolution do the down sampling and it be making the problem for the concatenation that's why down sampling is done on the transition layer. Because of this the size of each layer inside the block remains the same and it do not affect the concatenation and on the other hand down sampling is also done on the transition layer.

Dense block performs the work on the global average layer when the input reaches to the first dense block and this layer has size of 2*2. In last, it has a fully connected layer where it uses the SoftMax activation function and it provides the final output of the input data.

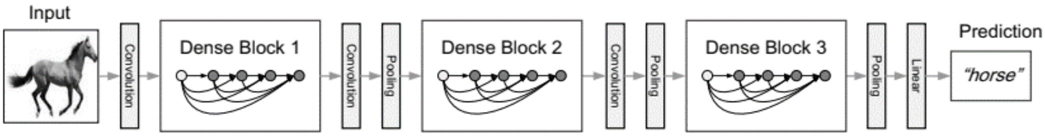


Figure 7 DenseNet for three dense blocks (Gao huang, 2017)

As DenseNet have a large number of parameter so this is why is comes with great computational cost. Direct connection of each layer has its influence the lessens the issue of the overfitting. (Shrivastav, 2022).

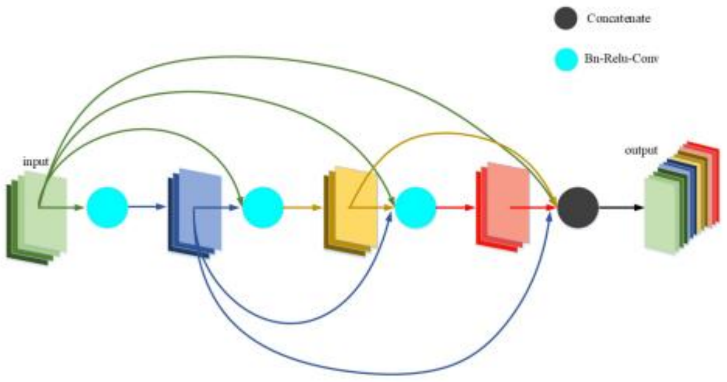
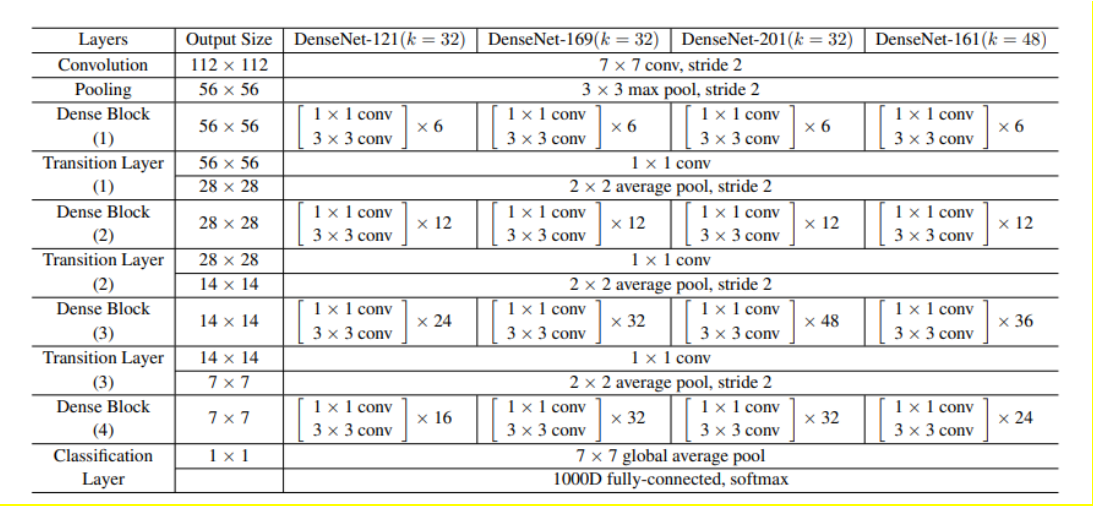


Figure 8 Densely concatenated convolution (Shrivastav, 2022)

b. Growth rate

Input features in certain number of layers can be find using K0 + Kx(1-1). In the following equation K0 shows the channel number in the layer.



Layers	Output Size	DenseNet-121($k = 32$)	DenseNet-169($k = 32$)	DenseNet-201($k = 32$)	DenseNet-161($k = 48$)
Convolution	112 × 112	7 × 7 conv, stride 2			
Pooling	56 × 56	3 × 3 max pool, stride 2			
Dense Block (1)	56 × 56	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$
Transition Layer (1)	56 × 56	1 × 1 conv			
	28 × 28	2 × 2 average pool, stride 2			
Dense Block (2)	28 × 28	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$
Transition Layer (2)	28 × 28	1 × 1 conv			
	14 × 14	2 × 2 average pool, stride 2			
Dense Block (3)	14 × 14	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 24$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 48$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 36$
Transition Layer (3)	14 × 14	1 × 1 conv			
	7 × 7	2 × 2 average pool, stride 2			
Dense Block (4)	7 × 7	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 16$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 24$
Classification Layer	1 × 1	7 × 7 global average pool			
		1000D fully-connected, softmax			

Figure 9 DenseNet architectures for (Huang et al.,2017).

Once written, the global state is accessible from anywhere on the network, unlike in conventional network topologies.

c. Advantages

- It minimizes the problem of vanishing gradient.
- It makes the feature propagation very strong.
- It is also good for the reusability.
- Decision tree also reduces the total number of parameters.

d. Disadvantages

- Due to the replication of feature map in each layer it causes to a duplication of a lot of data.
- It utilizes huge amount of memory because of the concatenation procedure

3. Xception

Xception consists of seventy-one deep layers. The conventional modules of the inception architecture are changed with the depth wise separable convolutions in the Xception. It is basically an upgrade in its predecessor Inception design. It has the same number of parameters, but they are efficiently used. (Chollet, 2017). Behind the idea of Xception the logic came all way through the 'Extreme Inception'. In Inception, in first stage original inputs are compressed by 1x1 and afterword from every input spaces later from each one of these input spaces, each of these depth spaces was subjected to a distinct set of filters. Where Xception, on the other hands is doing the same procedure but in reverse order. First, different filters are applied on the depth maps. When this step completed then it compresses the input space using 1x1 convolutions and it applies in the depth of the spaces.

This approach is almost similar to that of a depth-wise separable convolution. The difference between the Inception and Xception is the lack of non-linearity after the first operation. Xception do not use any non-linearity while in Inception the non-linearity (ReLu) comes after the completion of both process in the model. The foundation of the Xception model are 71 convolutional layers that serve as the feature extraction process. In Xception, the convolutional layers are grouped in the modules with the exception of the first and the last.

The architecture design and modification in Xception is very straight forward as compared to the inception V2 and V3 which are very complex to demonstrate. On the other hand, in Xception the layers are stacked together to form the xception and they are depth wise separated from each other. Xception designs consists of three flows i.e., entrance, middle and the exit flow in Figure 7. The data travels with the ratio of 299*299*3 as an input image in entry flow and leaves in 19*19*728 feature map. In middle flow it repeats itself for 8 time. Ultimately goes out of the system through the exit flow where it performs the logistic regression.

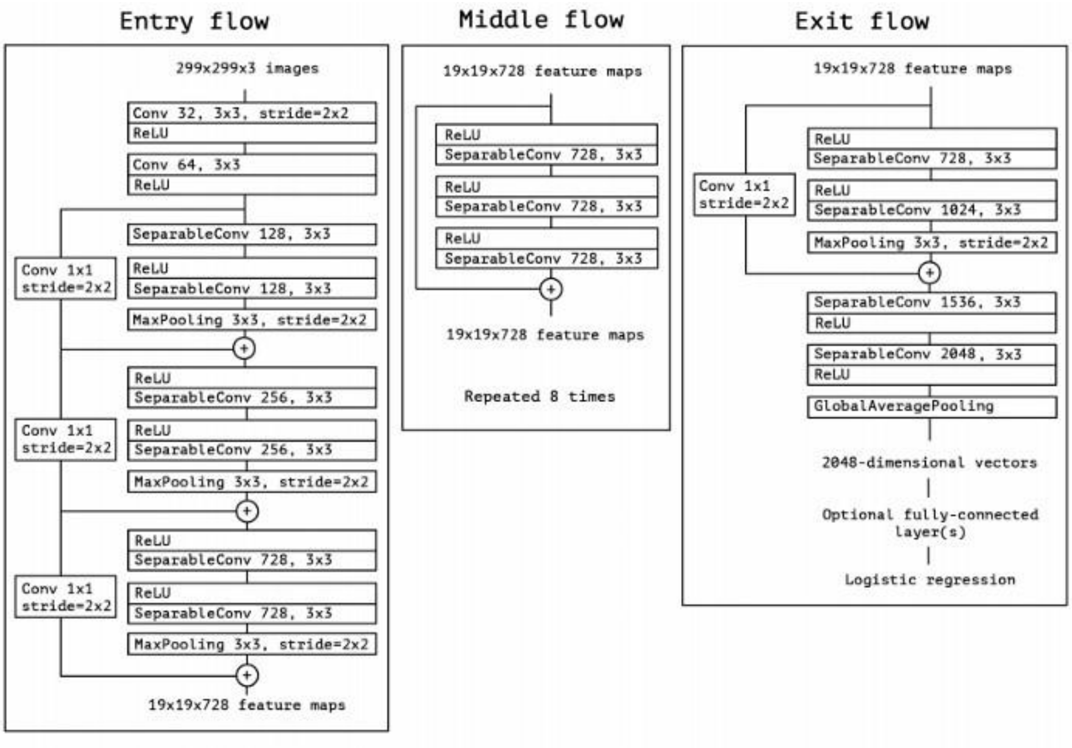


Figure 10 Xception architecture (Chollet,2017)

a. Depth-wise Separable Convolution

Architecture of the Xception is based on the dept-wise separable convolution. There are the details about the depth wise separable convolution.

Its name came from the idea as depth and spatial dimension of filter can be separated and that is why it is called depth wise separable. These convolutions are an alternative to traditional convolutions. Depth wise separable convolution separate into depth, spatial and in channel. The process of convolution is a very costly. For example, an input image includes a particular channel, denoted by the letter h. Also, it has a particular dimension labeled with V. We can use a convolution filter with a size of V=n*n to process it. For 1 kernel, it performs the procedures described in Equation (7), there M is the dimension that is obtained after convolution. This dimension is determined by the padding that is applied. Therefore, equation (8) is required to complete the equation for the depth of convolution.

$$M2\ x\ V2\ x\ H \qquad (7)$$

$$M2\ x\ V2\ x\ H\ x\ O \qquad (8)$$

The high cost of such procedures takes them to the development of convolutions which are depth-wise separable as a solution. Later, they are split into two phases known as the "Pointwise Convolution" and the "Depth-wise Convolution."

i. Depth-wise Convolution

In normal convolution, we convolve directly to the depth dimensions. In Depth-wise Convolution we use each filter channel only in one point of the input channel. It uses the depth filter for 2 dimensions which are basically used at every input level. This is the initial stage in the process, we apply a convolution with a size of d×d×1 and we perform it one channel at a time. This result then takes us to the point-wise convolution.

ii. Point-wise Convolution

Point wise convolution works on the approach of 1*1 kernel. This kernel iterates from a single point. A classical convolution is performed using pointwise convolution. It is convolutional operator that can output the features at each point in a cloud and the size of the operation is 1×1×O over the M×M×C volume. The number of total operations may be scaled back by a factor that is equivalent to 1/O.

4. EfficientNet

Conventionally most of the CNN works on making deep neural networks which are good for increasing the accuracy results but they stop working at a certain point giving birth to the problem is vanishing gradient. Google created EfficientNet, a model whose function is to classify images. EfficientNet do not only do the scaling for depth, but it also scales the width and resolution. Increase in the resolution provide more details of an image while width helps to increase the total number of feature maps. (Sarkar, 2021). The cost of the resources used to create convolutional neural networks is fixed. Later, as more resources are available, these networks are scaled up to produce improved accuracy. By adding extra layers to the initial model, a ResNet 18 model may be expanded up to a ResNet 200 model. This scaling strategy has typically contributed to greater accuracy on the majority of benchmarking datasets. However, the traditional methods of model scaling are quite sporadic. Some models have depth scaling, while others have width scaling. To achieve better outcomes, some models merely consume photos with a higher resolution. This method of arbitrarily scaling models necessitates human adjustment and several man-hours, frequently yielding little or no performance increase.

a. Compound model scaling

Performing the scaling on resolution gives many complex feature and fine patterns. Similarly, width scaling gives more channel. In order to handle these, it is also required to increase the number of layers or the depth of the neural network. But the most important thing is that they should be scaled but up to which content. As, they should not be scaled too much. This is the reason that the author represents the idea of compound scaling. Where, the work of author suggests scaling everything in a balanced manner (Tan, 2019). Compound scaling details are mentioned below in figure (11).

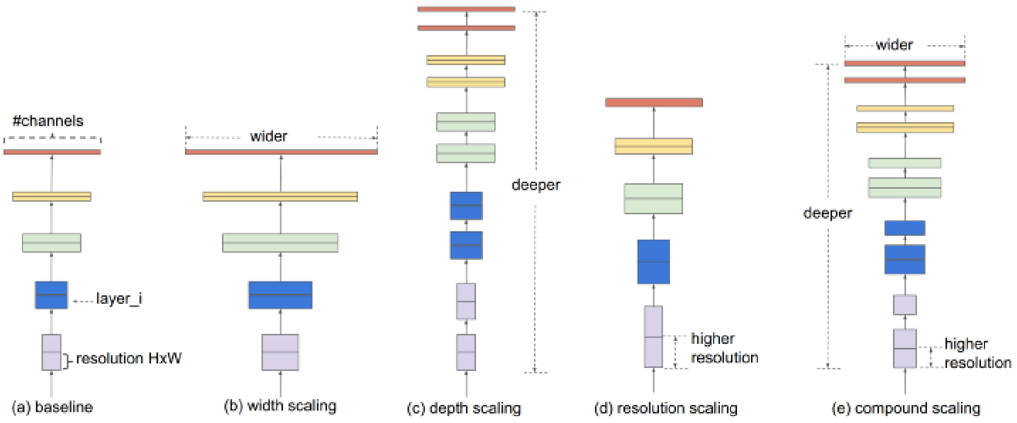


Figure 11 Different scaling methods vs. Compound scaling (Tan, 2019)

The concept behind the compound scaling approach is to scale with a constant ratio to achieve a balance scaling. The equations below demonstrate how it is mathematically accomplished.

$$depth\ D = \alpha^{\emptyset}, width\ w = \beta^{\emptyset}, resolution\ r = \gamma^{\emptyset} \qquad (9)$$

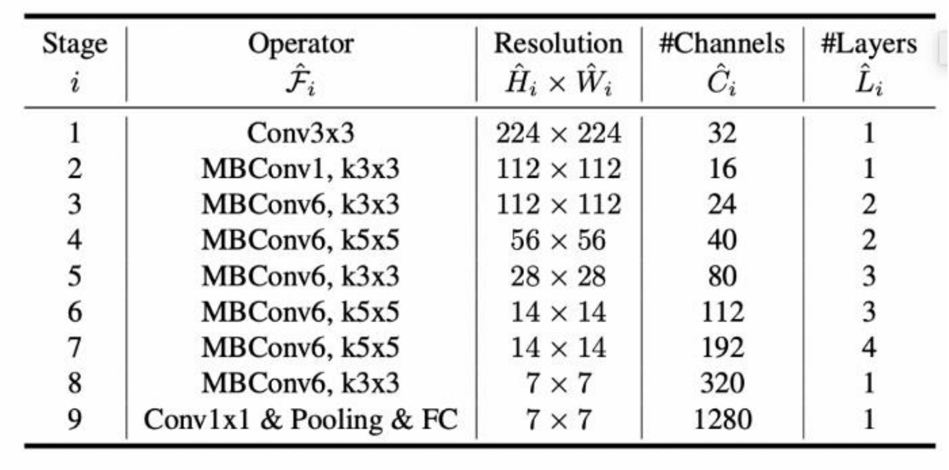
$$such\ that\ \alpha, \beta^2, \gamma^2 \approx 2$$

$$\alpha \geq 1, \beta \geq 1, \gamma \geq 1$$

The intuition for the networks is, when the input image is large then it requires deeper layered network to obtain the fine patterns and getting the features. This technique provides with more accuracy for the model.

b. Architecture

EfficientNet is based on its baseline model. Its baseline model was not invented by the humans rather it was developed by the NAS. Every model in this family starts its scaling from the baseline model of ‘EfficientNet B0’. it is clear how the various models differ from one another. They also steadily raised the number of sub-blocks.



Stage i	Operator $\hat{\mathcal{F}}_i$	Resolution $\hat{H}_i \times \hat{W}_i$	#Channels $\hat{C}_i$	#Layers $\hat{L}_i$
1	Conv3x3	224×224	32	1
2	MBConv1, k3x3	112×112	16	1
3	MBConv6, k3x3	112×112	24	2
4	MBConv6, k5x5	56×56	40	2
5	MBConv6, k3x3	28×28	80	3
6	MBConv6, k5x5	14×14	112	3
7	MBConv6, k5x5	14×14	192	4
8	MBConv6, k3x3	7×7	320	1
9	Conv1x1 & Pooling & FC	7×7	1280	1

Figure 12 Kernel Size, resolution, channels, and no of layers information (Tan,2019)

c. EfficientNet-B0

This NAS developed baseline model get a training using around 1 million images. It gets an input in a fixed size of 224*224. This model can categorize one thousand various categories.

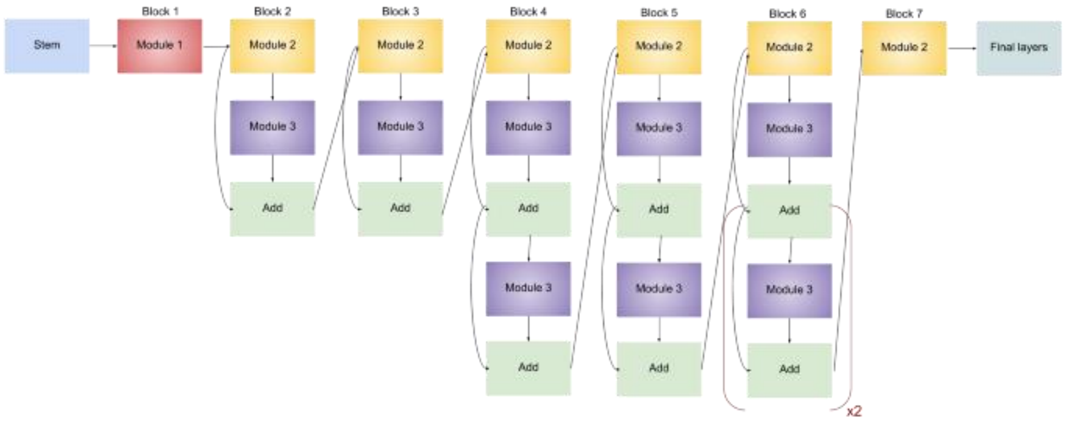
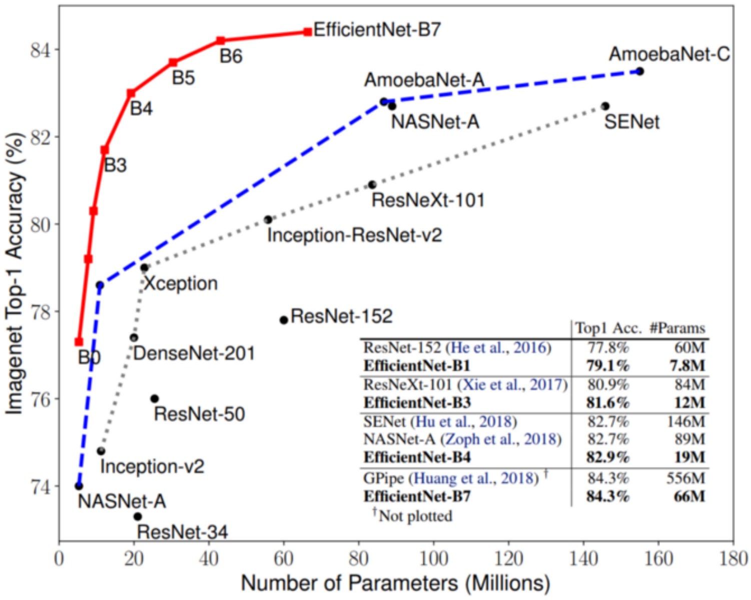


Figure 13 EfficientNet-B0 architecture (Tan,2019)

c. Performance

On the x-axis we have parameter in millions. While in Y-axis we take the score for the accuracy. In the beginning of the performance graph as it is mentioned in figure (10), Inception utilize around ten million parameters while it gives an accuracy of around 74.5 %. On the other hand, B3 network which also utilizes almost the same parameters gives an accuracy of 80%. Performance graph shows that EfficientNet-B7 outperforms the rest of the other models using a parameter of around 60 million. While ResNet, SENet etc. take a greater number of parameters but it shows less accuracy.(Sarkar, 2021).



	Top1 Acc.	#Params
ResNet-152 (He et al., 2016)	77.8%	60M
EfficientNet-B1	**79.1%**	**7.8M**
ResNeXt-101 (Xie et al., 2017)	80.9%	84M
EfficientNet-B3	**81.6%**	**12M**
SENet (Hu et al., 2018)	82.7%	146M
NASNet-A (Zoph et al., 2018)	82.7%	89M
EfficientNet-B4	**82.9%**	**19M**
GPipe (Huang et al., 2018) †	84.3%	556M
EfficientNet-B7	**84.3%**	**66M**

†Not plotted

Figure 14 Performance and size for the image data set (Tan,2019)

The model was also seen to offer improved Class Activation Maps (CAM), which put more of an emphasis on the pertinent areas with more object information, opening the door to a more understandable model.

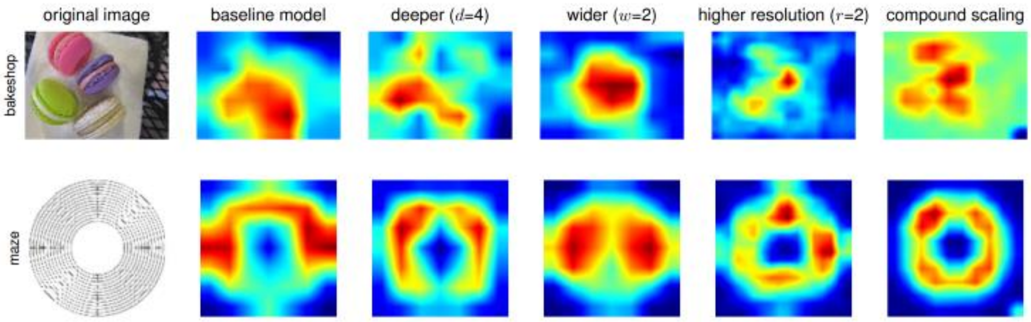
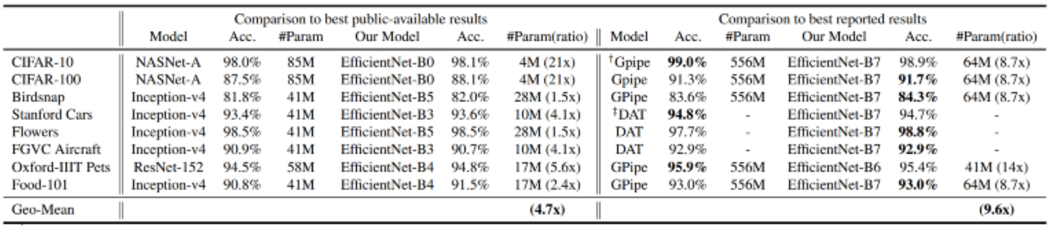


Figure 15 Compound scaling revelations (Tan,2019)

On the majority of benchmarking datasets, EfficientNet's performance surpasses that of all prior CNN designs. Other CNN designs may be scaled well with the usage of compound scaling approach as well.



	Comparison to best public-available results						Comparison to best reported results					
	Model	Acc.	#Param	Our Model	Acc.	#Param(ratio)	Model	Acc.	#Param	Our Model	Acc.	#Param(ratio)
CIFAR-10	NASNet-A	98.0%	85M	EfficientNet-B0	98.1%	4M (21x)	†Gpipe	**99.0%**	556M	EfficientNet-B7	98.9%	64M (8.7x)
CIFAR-100	NASNet-A	87.5%	85M	EfficientNet-B0	88.1%	4M (21x)	Gpipe	91.3%	556M	EfficientNet-B7	**91.7%**	64M (8.7x)
Birdsnap	Inception-v4	81.8%	41M	EfficientNet-B5	82.0%	28M (1.5x)	GPipe	83.6%	556M	EfficientNet-B7	**84.3%**	64M (8.7x)
Stanford Cars	Inception-v4	93.4%	41M	EfficientNet-B3	93.6%	10M (4.1x)	‡DAT	**94.8%**	-	EfficientNet-B7	94.7%	-
Flowers	Inception-v4	98.5%	41M	EfficientNet-B5	98.5%	28M (1.5x)	DAT	97.7%	-	EfficientNet-B7	**98.8%**	-
FGVC Aircraft	Inception-v4	90.9%	41M	EfficientNet-B3	90.7%	10M (4.1x)	DAT	92.9%	-	EfficientNet-B7	**92.9%**	-
Oxford-IIIT Pets	ResNet-152	94.5%	58M	EfficientNet-B4	94.8%	17M (5.6x)	GPipe	**95.9%**	556M	EfficientNet-B6	95.4%	41M (14x)
Food-101	Inception-v4	90.8%	41M	EfficientNet-B4	91.5%	17M (2.4x)	GPipe	93.0%	556M	EfficientNet-B7	**93.0%**	64M (8.7x)
Geo-Mean						**(4.7x)**						**(9.6x)**

Figure 16 EfficientNet Performance Results on Transfer learning dataset (Tan,2019)

5. Inception

A good deep neural network can be made if add deep layers in the network (Szegedy, 2015). Aforementioned method has two drawbacks: it requires more computation and may result in overfitting if a neural network has too many layers, especially If there aren't enough labelled training data. Inception networks were developed with the goal of enhancing a deep neural network's functionality while effectively utilizing computing resources. Inception network consists of a deep learning network which has number of different modules in the network. Each revision represents an incremental improvement over the one before it. We can construct unique classification from that by adding some improvements.

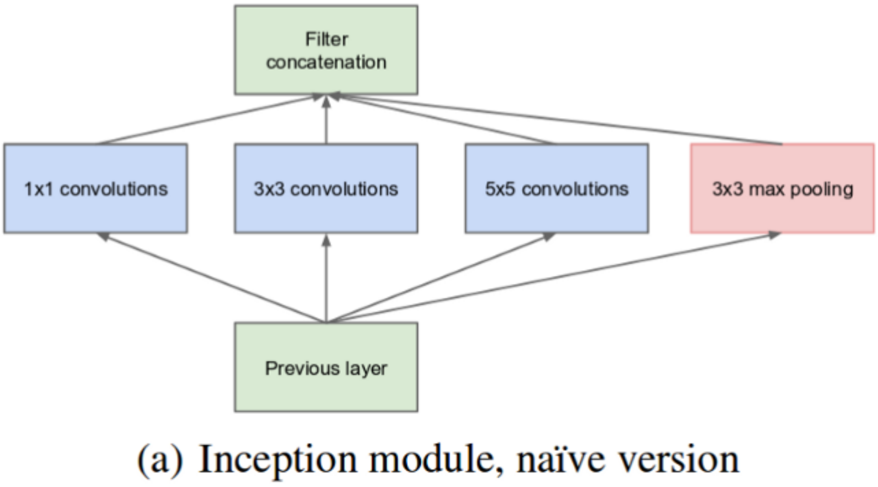


Figure 17 Inception module, naïve version (Moniruzzaman M, 2019)

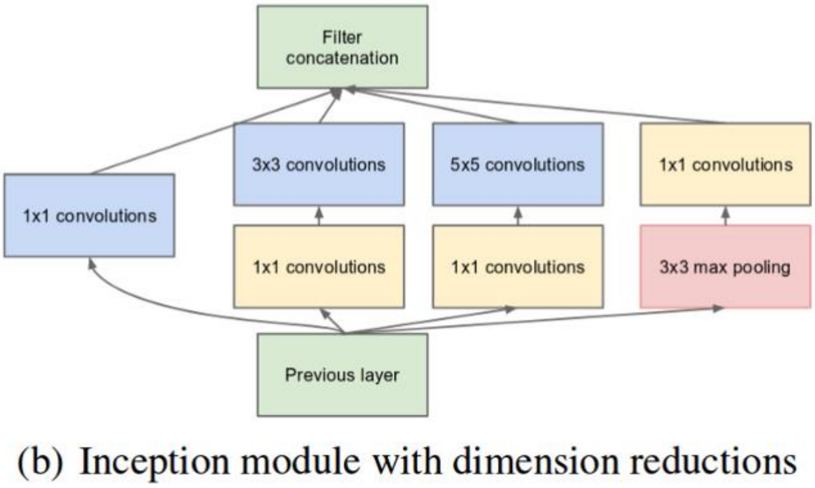


Figure 18 Reduction of the dimensions (Moniruzzaman M, 2019)

a. Inception v1

The size of important portions of the picture might vary dramatically. For instance, the images below illustrate two possible dog-related images. Each picture shows a different region that the dog is in (Ghantiwala, 2022).

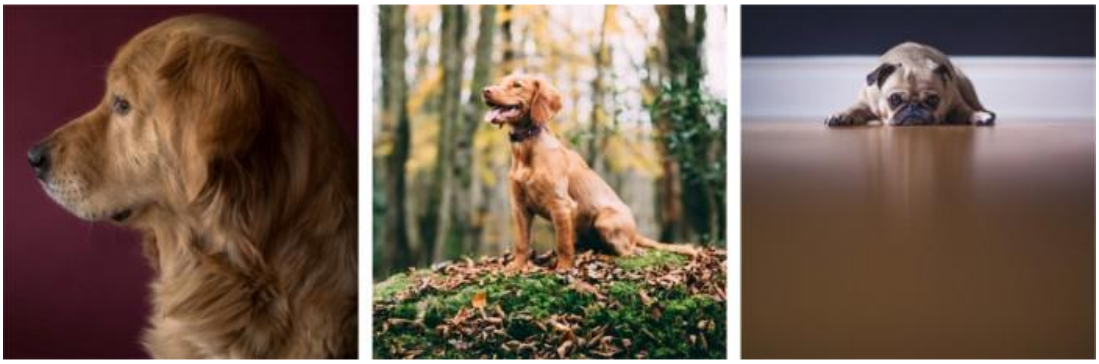


Figure 19 Different ratio of object in images (Ghantiwala, 2022)

As mentioned above, in different picture there are different revelations of the object. In one picture the desired object is seen on the most of the screen but on the other hand in one image the object is seen the half of the screen. In the last picture, object is seen on the small portion of the image, Inception module uses convolution of different sizes, The authors depict the use of dimension reduction in the first module. As, the computation for the networks are quite costly. It is very simple. Model will use the 1*1 convolution before using the 3*3 and 5*5 convolution. It can reduce the number of parameters in a very efficient way. Most important thing is that the size of convolution will decrease after the Max pool layer. When, there is a small information it will need to have small filters but when the information is large there is also a need to have a bigger filter.

b. Inception v2

Diminution of the representational bottleneck the assumption was that neural networks operate more effectively when convolutions don't significantly change the input's dimensions. A "representational bottleneck" is a loss of information that can occur when the dimensions are reduced too much. Convolutions may be made more computationally efficient by using clever factorization techniques. To increase computing speed, factorize the 5*5 convolutions with two 3*3 convolution layers. This factorization helps to increase for computing. It reduces the size of parameter hence cost and time saving.

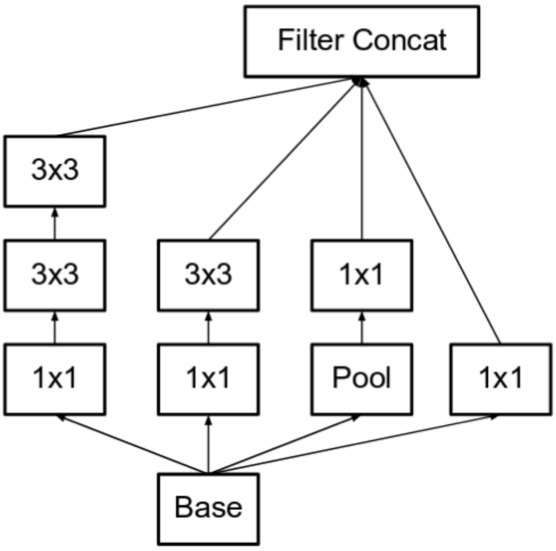


Figure 20 5x5 convolution (Szegedy,2015)

They combine 1xn and nx1 convolutions to factorize n*n filter size convolutions. For example, performing a 1x3 convolution first and after that applying a 3x1 convolution to the output value, produces a 3x3 convolution. They discovered that their approach was 33% less expensive than a single 3x3 convolution. In the picture below, this is demonstrated.

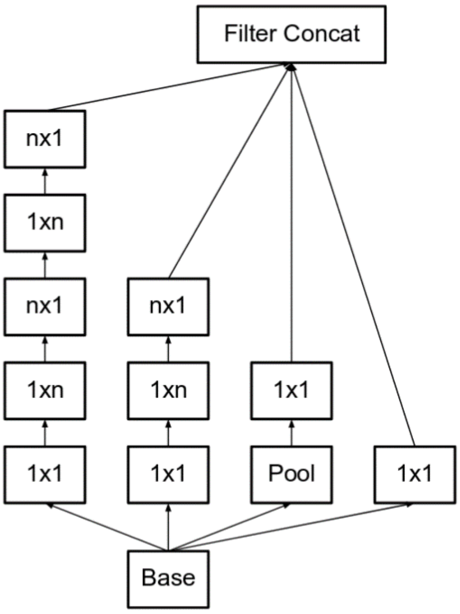


Figure 21 Layers architecture (Szegedy,2015)

In order to eliminate the representational bottleneck, it increases the filter in module. Module would lose information if it were made deeper since the dimensions would be drastically reduced. In the picture below, this is demonstrated.

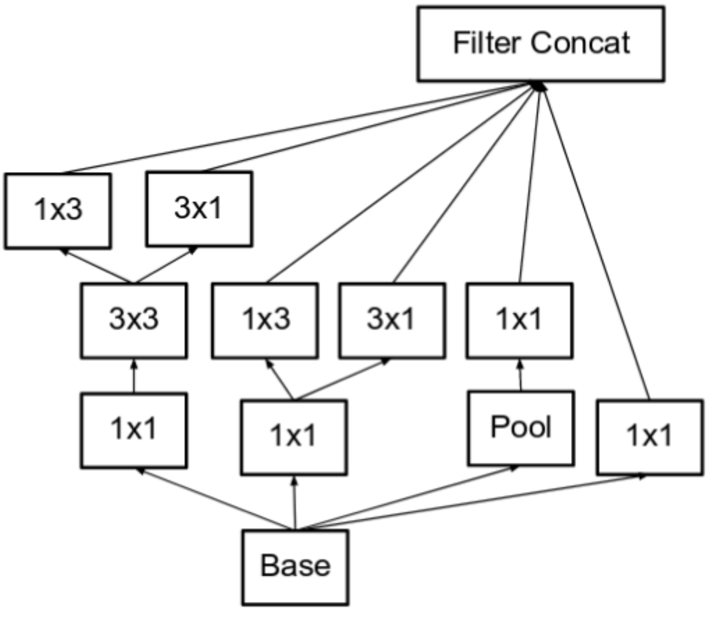


Figure 22 Making the inception module wider (Szegedy,2015)

c. Inception v3

When accuracies were approaching saturation, the auxiliary classifiers didn't contribute significantly. Normally when they use the process of Dropout, they act as regularizes (Alake, 2020). It was necessary to look at ways to enhance Inception v2 without significantly altering the modules. It uses their different factorization techniques to make it more efficient. First, it will factorize into small convolutions replacing the lower convolutions before the higher one. Second, it uses an asymmetric convolution where it replaces 3*3 into resultant 1*3 and 3*1. It helps to reduce the number of parameters indeed it's a goal to reach to the minimum number of parameters. In the end it used another asymmetric 7*3 convolution to reduce more parameters. Inception v3 takes an input in a fixed size (299*299*3).

6. Random Forest

It is a form of decision tree where it selects the random results obtain from different decision tree to show the classification. It is versatile as we use it for both classification and also for the regression problem. In Random Forest, a dataset is splitted into sub-sets and different decision trees are originated from the datasets. Then, it uses the bagging technique and picks the result which shows the maximum vote from the decision tree and use it as a final result of the prediction. Let say, we have a data set with different apple and peach. Three decision trees are originated from this dataset. Now, first decision tree predicts it as apple. Second, gives the value of a peach. Where, the third decision tree shows us an apple. Now, random forest will use the maximum vote from the decision tree. As a result, it classifies the problem as the apple.

Building blocks for the random forest algorithms are decision tree. There are various terminologies in random forest. At root node, the training dataset is used, and it divides into sub nodes where different decision trees give birth. In random forest, splitting is also used. For this, Random Forest uses the Gini impurity and the entropy to decide the best optimization of the best split. There are decision nodes as well. Decision nodes are very important as it will take to the original prediction value. The most important is the lead node, from where there cannot be no decisions make. It is the end point for the decisions.

It is very important to understand the working of the random forest. The best split in random forest is chosen by the Gini impurity, entropy and the information gain. Here, data is splitted into different data splits. There random samples are used and then it used a technique of replacement of random sampling. The result that it obtains from the random sampling becomes the bootstrap aggregation or bagging.

Splitting in random forest based on different things. Entropy is a metric that is used to find the purity or impurity of a dataset. The purity and impurity lead us to the correct node to proceed with our process of reaching the final node. (Saini, 2021). This metric is used to check especially the impurity in a node. Here, important thing to understand the is the impurity. Impurity predicts that how much randomness we have in our data. To build a decision tree means to find the best attributes which can yield maximum information gain. When we have different attributes then the IG decides that which decision node should be chosen. It is gained according to the reduced impurity in a particular node. Information gain depicts the correct value after splitting the root node and it calculates the value as an average of the entropy before and after the splitting of the dataset. It is used in the decision tree during the iterations. To identify how the characteristics of a data collection should divide nodes to create the tree, decision trees are constructed using the Gini Impurity measurement (Chauhan, 2022). In more detail, the Gini Impurity of a data set is a value between 0-0.5 that represents the probability that fresh, random data will be incorrectly categorized if a random class label is assigned to it in accordance with the data collection's class distribution.

It works on the decision trees. A dataset is divided into subsets where different decision trees are applied on it. This data separates intro various branches by the decision tree. This process keeps doing the iteration until it reaches to a leaf node. Figure 19 shows the process in a minimalist way.

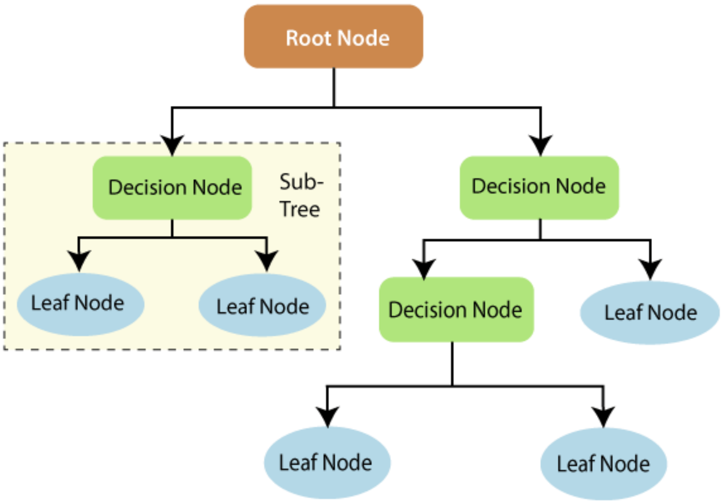


Figure 23 Leaf nodes (Mbaabu, 2020)

a. Advantages

As the random forest uses the ensemble technique or bootstrap aggregation it is useful in reducing the overfitting of the data. It has a low variance as it use multiple decision trees. It works on the approach of rule base hence normalization is also not required. It gives the good accuracy for the prediction as it chooses the mostly voted prediction from different decision trees. It is useful for both classification and aggression problems. It can perform better for continuous and categorical data. Random forest can also perform better for the larger datasets.

b. Disadvantages

When a technique is coming with a lot of advantages then it may also come with some disadvantages as well. As, it works on the basis of the results generated by different decision trees. So, it needs more time for the training. Especially when we have a bigger dataset, then it will generate hundreds of decision trees and it will take a lot of time. Interpretation also becomes difficult when it needs to interpreted from hundreds of the decision trees. It also utilizes a lot of memory. Along with that, it also gives a lot of computational cost.

b. Random forest classification

Classification is done using the decision tree. It uses the bagging approach where it utilizes the ensemble process. Training data is fed to many decision trees. Dataset contains different type of attributes and characteristics. Each decision tree makes its own tree and give the prediction in the last on the leaf node. So, when different decision trees give different values then the random forest get the prediction from the highest voting. The architecture is simple. It is mentioned below (Figure 24).

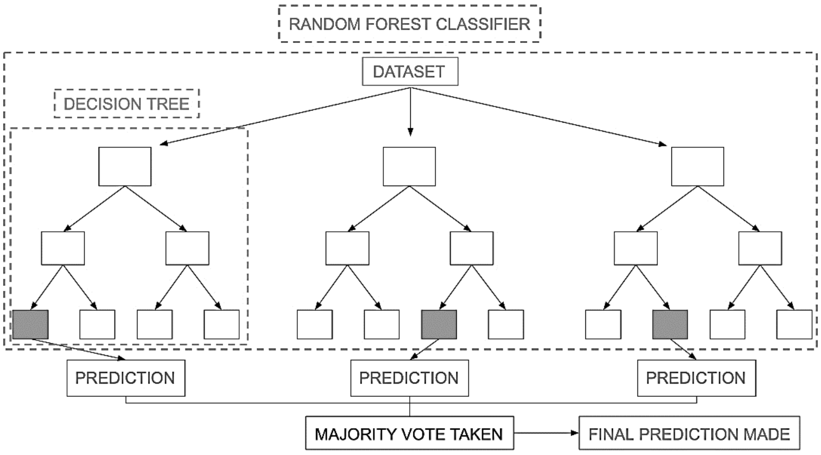


Figure 24 Random Forest classifier (Mbaabu,2020)

There is an example where the training dataset consists of a basket of fruit which contains apples, bananas and other fruits in it. Random forest divides this dataset into different decision trees. When it applies the bagging technique and gets the data based on maximum vote. It gets the final class or result as an apple.

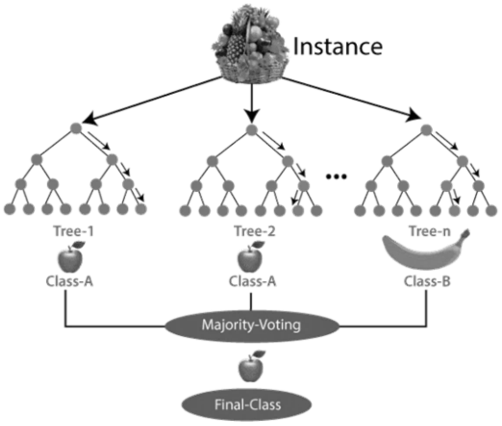


Figure 25 Random Forest architecture (Mbaabu,2020)

7. Support vector machine (SVM)

SVM is one of the popular algorithms now a days which is useful for both classification and regression problem. Mostly it is useful in bioinformatics and image-classification problems. It debuted in the early 1960s before being improvised in the 1990s. This algorithm needs to have a supervised learning. It gets a fame because of its efficient results. It can be used for the classification, regression, and early detection. This algorithm is based on the hyperplanes. Good thing about this algorithm is that it can also be used to make non- linear classification. It uses hyperplanes to segregate the classes in efficient way and then right highest point of hyperplane from the nearest data point. Sometimes, hyperplanes can also be not efficient. There, SVM use kernels to transform the input to highest dimensional spaces. It helps to segregate the spaces.

a. SVM Kernels

SVM kernels are useful for increasing low-dimensional datasets dimensionality so that the data may be more easily segmented. By applying kernel methods to add new dimensions, it converts the indivisible problems into separable problems. Kernels are always used to implement SVM. Making an accurate classifier with it is beneficial. SVM have a variety of kernels.

i. Linear kernel

When used as a dot product between any two observations, it can be beneficial. The sum of the multiplications of each pair given the input value is the product of the two vectors.

ii. Polynomial kernel

It is also a form of linear kernel but it can also differentiate between the curved and non- linearity in the spaces.

iii. Radial Basis function (RBFK):

It is the mostly used kernel by the SVM classification. It is helpful in finding the map space in infinite dimension.

b. Working of SVM

It is fed to the data where the classification problems need to be done. There, SVM explores the data first. It checks that how many labels it contains and which variables serve as the target? Then it splits the data into train and test data. After splitting the data SVM generates a model for the classification.

c. Advantages and Disadvantages

For conventional algorithms the dimensions need to adjusted for the algorithm but SVM is useful as it at can be adjusted automatically for the higher dimensions. It also does the regularization, as many algorithms falls into overfitting but in SVM it adjusts itself to avoid the overfitting and biased problems. It is useful when the dimensions are higher as compared to the no of specimens. SVM is memory systematic.

As compared to other algorithms like Random Forest, It cannot good for large datasets. It can also not work when different classes overlap each other. SVM working is done by placing the data points below and above the hyperlane so it does not have the probability-based classification. SVM draws a boundary which separate the number of dimensions into classes. It is useful for the classification. (Gandhi, 2018). SVM finds the most separation from two classes to categorize the data points. (Ray, 2017).

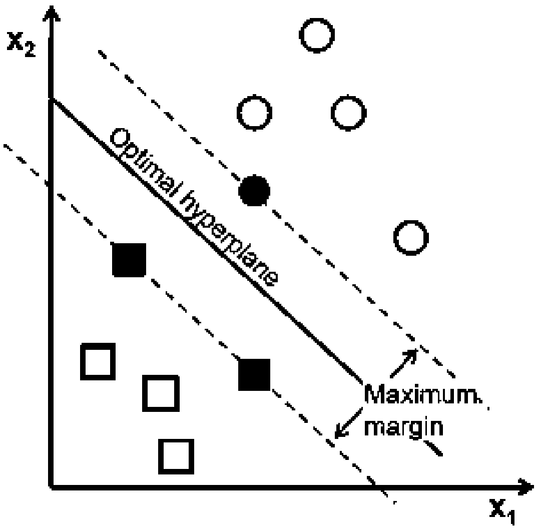


Figure 26 Possible hyperplanes (Gandhi,2018)

d. Hyperplanes and support vectors

Hyper lanes came into existence according to the number of features that we have. We will have a two-dimensional decision boundary when we have two features. It generates three dimensional hyper lanes when we have 3 features and it go more and more with the increase in the feature. Figure (27) shows the complete details.

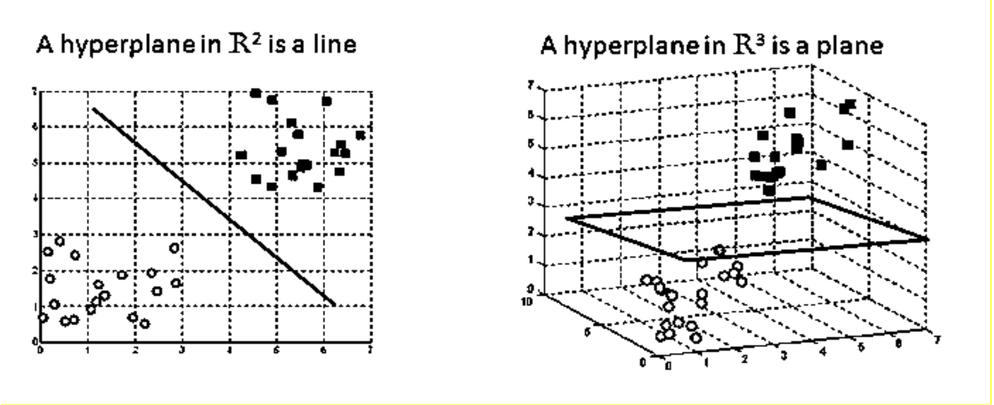


Figure 27 Hyperplanes in 2D and 3D feature space (Gandhi,2018)

In the graph, hyperplanes also come with the assistance of the support vectors. It is the closest data points to the hyper lanes. It is the base of an SVM. If any support vector is deleted, it will directly impact the hyperplane position.

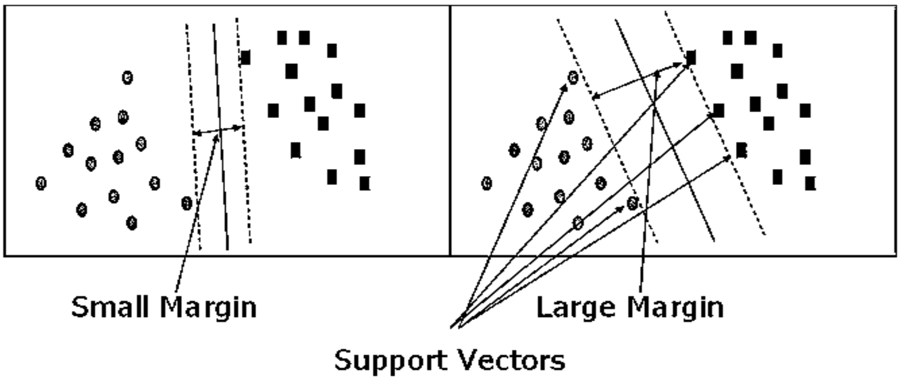


Figure 28 Support vectors (Gandhi,2018)

SVM method maximize the distance between the hyperlanes and data points. To increase the distance, hinged function is used.

$$C(x, y, f(x)) \ = \ (1 - y \ * \ f(x))_+ \qquad (10)$$

The loss function for SVM is as below:

$$min_w \lambda||w^2|| \ + \ \sum_{i=1}^{n}(1 - y_i(x_i, w))_+ \qquad (11)$$

To get the gradients, consider the partial derivates where the modify weights use gradients. These are as below:

$$\frac{\delta}{\delta w_k}\lambda||w^2|| = 2\ \lambda w_k \qquad (12)$$

$$\frac{\delta}{\delta w_k}(1 - y_i(x_i, w))_+ = \begin{cases} 0, & if\ y_i(x_i, w) \ >= \ 1 \\ -y_i x_i k, & else \end{cases}$$

When it accurately gives the prediction of class for data & there is no misclassification. Gradient update is as below:

$$w \ = \ w - \alpha\, .\, (2\ \lambda \mathrm{w}) \qquad (13)$$

When a data point is misclassified the regularization parameter & loss are both used for execution of a gradient update.

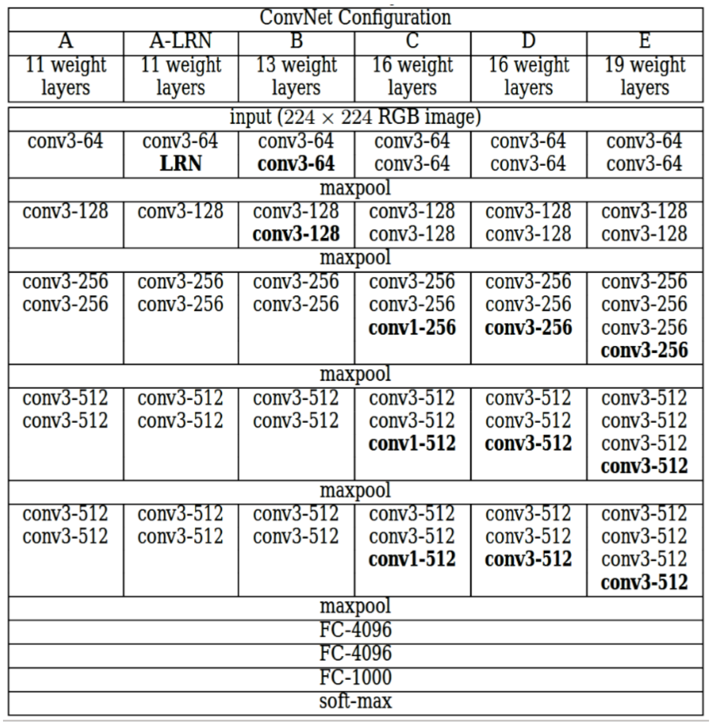
$$w \ = \ w + \alpha\, .\, (y_i . x_i - 2\ \lambda \mathrm{w} \qquad (14)$$

8. VGG19

It is a type of VGG which have 19 layers deep model which is based on CNN and it is called as VGG-19. It has total 16 layers in which 13 are convolution layers and three of them are fully connected. As a pre-processing model, CNN is utilized. The network depth of the VGG-19 is more maximized if we compare it with the previous versions of VGG and conventional CNN. It switches between numerous convolutional layers and non-linear activation layers in place of using just one convolutional layer. Better picture feature extraction, down sampling via Maxpooling, and activation function modification are all possible with the layer structure (Xiao).

a. VGG19 architecture

The input for this network comes with a concrete size of (224 * 224) RGB picture, which shows that the matrix has a shape of (224,224,3). It depicts that during the training phase the whole thing that was calculated as a pre-processing part was the value of RGB. In the architecture of VGG-19, the kernel was used of the size (3 * 3) in size and it use the stride as 1 pixel. To keep the spatial resolution of the image, spatial padding was also utilized. Stride 2 was also utilized in order to conduct the max-pooling on 2 * 2-pixel ratio window. After this process, the next process is to use Rectified Linear Unit, which contains non-linearity which can increase the ability of the model to categorize data and shorten computation time. Functions this far outperformed them. The column E is for the VGG19. Following images explains the complete architecture in detail.



ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Figure 29 Variants of VGG models (Simonyan, 2015)

IV. EXPERIMENT AND EVALUATION

In light of the fact that this study focuses on a method for detecting a plant disease based on the leaf condition of the certain plant, it is essential to assess processing and classification performance. Computers using an Intel(R) Core (TM) i5, CPU 3.0GHz and Apple M1 Chip processor are used throughout the training and testing processes. The network was built using the TensorFlow framework, and 9 separate Keras applications were used to fine-tune it using different datasets containing the images of different plants. A large dataset was built consisting of 15,852 images. In this dataset, each class consists of approximately 450 images.

The used metrics are:

- Accuracy: It provides the accurate results that we obtain from different number of observations from our model that either they classified as positive or negative.
- Precision: It shows the percentage of instances that has been referred to the correct category.
- Recall: Using all of the valid cases in the dataset, it is used to calculate the overall number of accurate positive class predictions.
- F1-Score: It come up with the integration of accuracy and recall into a single metric which is achieved from the computation of the harmonic median between the accuracy and recall.

Table 4 Performance indicators of different models

Model	**F1-score (%)**	**Accuracy (%)**	**Recall (%)**	**Precision (%)**
DenseNet121	88.00	87.95	81.10	84.70
EfficientNetB0	86.90	86.90	81.10	81.00
InceptionV3	87.80	87.78	81.80	83.40
VGG19	83.70	83.68	75.10	77.90
Xception	89.50	89.53	84.10	85.20

Random Forest	51.52	65.34	52.8	55.23
SVM	40.82	56.83	43.00	46.87
Decision Tree	40.19	48.02	40.45	40.05

In (Table 4) The trained model that we have been using to analyze the images of plants with diseases is providing flawless results. We can see that for plant disease classification the most effective results from pre-trained models come from the Xception model, heaving the highest accuracy and from classical machine learning algorithms the Random Forest generated the good results than SVM and Decision Tree. Following the completion of training phase, the accuracy and loss were determined by applying each and every image from the test dataset to each and every iteration. Figures 30,31,32, 33 and 34 provide a visualization of the training iterations increased for each of our pre-trained models.

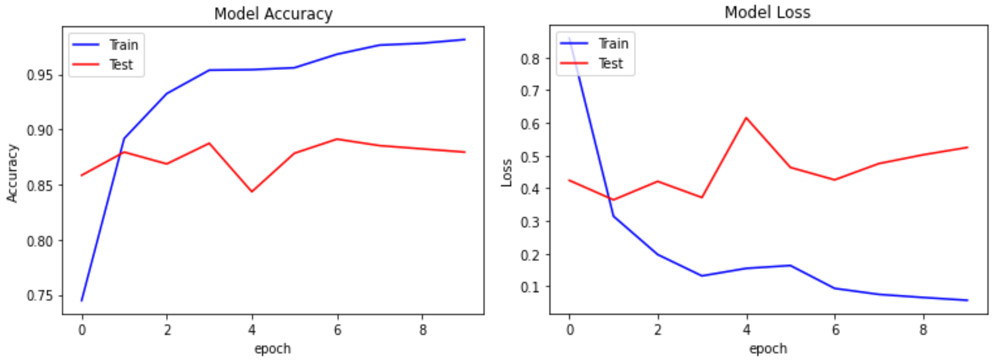


Figure 30 Visualization of DenseNet121 model for plant images

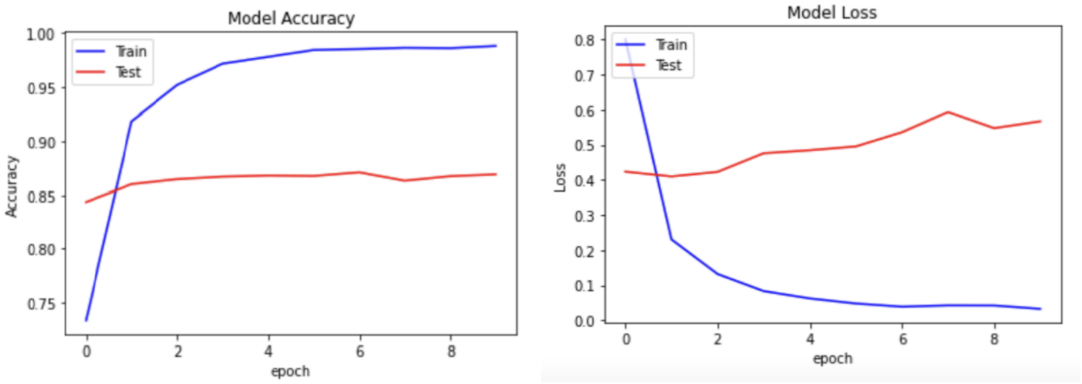


Figure 31 Visualization of EfficientNetB0 model for plants images.

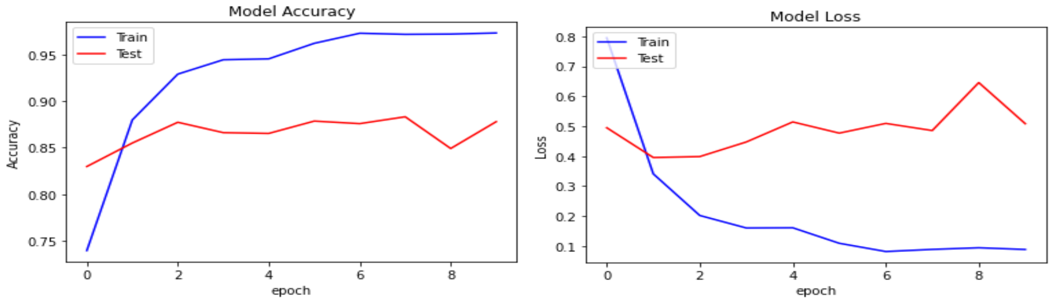


Figure 32 Visualization of Inception model for plants images.

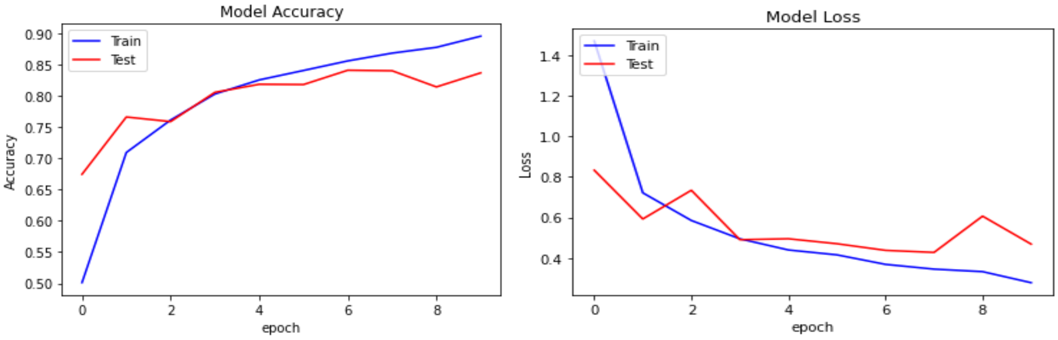


Figure 33 Visualization of VGG19 model for plants images.

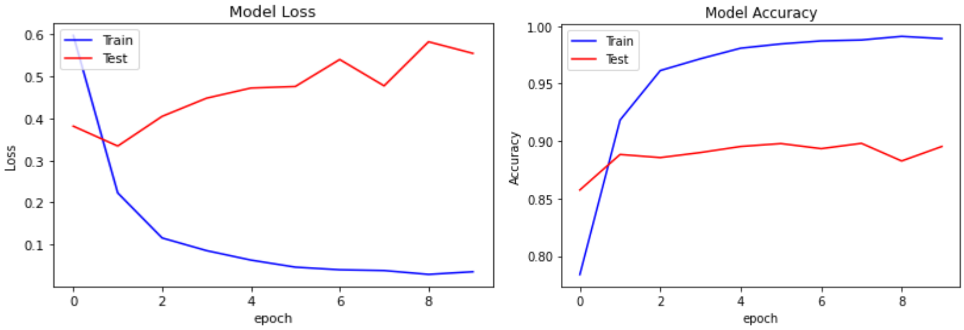


Figure 34 Visualization of Xception model for plants images.

V. CONCLUSION

Plants are recognized as a vital source of living. Detecting a plant disease especially in the early stages not even give the benefit avoiding many expenses on pesticide sprays and human efforts but it also helps to yield more and healthy crops. However, because of the complexity of the work, plant disease detection using images in different light conditions and area of the disease is a very complex task. According to the prior knowledge, there is a very little research on different plants and their diseases. Using the research done so far, we can now begin to create a system that will allow us to categorize the leaves of diseased plants to identify those plants. This search helped us to evaluate different machine learning algorithms and modern deep learning methods to diagnose the condition of the disease of plants.

In this work different datasets were combined to gather number of plant species with different number of healthy and diseased plant. It was then arranged to a fix ratio. Then models get training from these images. Our worked shows that the modern deep learning algorithms perform better than classical machine learning techniques. Especially, Xception gives the best classification accuracy apart from others. This work reached to a conclusion that this method of classification for the real-time illness identification of plants, categorization for plant diseases can be employed. It can also be implemented in different ways I.e., Agricultural drone or smart phones. Even in the future more work can be done by using fine-tuning and making specialized deep network architecture.

VI. REFERENCES

ARTICLES

AHMAD, A., DHARMENDRA, S., & Gamal, A. (2023). “A Survey on Using Deep Learning Techniques for Plant Disease Diagnosis and Recommendations for Development of Appropriate Tools”, **Smart Agricultural Technology3,** 100083. https://doi.org/10.1016/j.atech.2022.100083

ALI, H., LALI, M.I., NAWAZ, M.Z., SHARIF, M., & SALEEM, B.A.(2017). “Symptom based automated detection of citrus diseases using color histogram and textural descriptors”, **Computers and Electronics in Agriculture, Volume 138**, pp. 92-104

ARSENOVIS, M., KARANOVIC, M., SLADOJEVIC, S., ANDERLA, A., & STEFANOVIC, D.(2019). “Solving Current Limitations of Deep Learning Based Approaches for Plant Disease Detection", ***Symmetry,*** 11(7):939. https://doi.org/10.3390/sym11070939

BHANGE, M., HINGOLIWALA, H.A. (2015). “Smart Farming: Pomegranate Disease Detection Using Image Processing”, **Second International Symposium on Computer Vision and the Internet,** Volume 58, pp. 280-288, 2015.

CHOHAN, M., KHAN, A., CHOHAN, R., KATPAR,S.H., Mahar, M.S. “Plant Disease Detection using Deep Learning”, **International Journal of Recent Technology and Engineering(IJRTE), 2020.**

CholleT, F. (2017). Xception: “Deep Learning with Depthwise Separable Convolutions”, **Computer Vision and Pattern Recognition**. https://doi.org/10.48550/arXiv.1610.02357

CHOWDHURY, M., RAHMAN, T., KHANDAKAR, A., AYARI, M.A., KHAN, A.U., KHAN, M.S., AL-EMADI, N., REAZ,M.B.I., ISLAM, M.T., & ALI, S.H.M. (2021). “Automatic and Reliable Leaf Disease Detection Using Deep Learning Techniques”, ***AgriEngineering,*** 3(2):294-312.

https://doi.org/10.3390/agriengineering3020020

ELHASSOUNY, A., & FLORENTIN, S. (2019). “Smart Mobile Application to Recognize Tomato Leaf Diseases Using Convolutional Neural Networks”, **Proceedings of 2019 International Conference of Computer Science and Renewable Energies, ICCSRE 2019,** 1–4. https://doi.org/10.1109/ICCSRE.2019.8807737

FERENTINOS,K. (2018). “Deep learning models for plant disease detection and diagnosis”, **Computers and Electronics in agriculture, 2018**. 145 311-318

HUANG, G., LIU, Z., MAATEN, L.V.D., KILIAN,Q, WEINBERGER, K.Q. (2016). “Densely connected convlutional networks”, **Arxiv.** https://doi.org/10.48550/arXiv.1608.06993

JAVIDAN, S.M., AHMAD, B., KEYVAN, A., & YIANNIS, A. (2023). “Diagnosis of Grape Leaf Diseases Using Automatic K-Means Clustering and Machine Learning”, **Smart Agricultural Technology 3, 100081.** https://doi.org/10.1016/j.atech.2022.100081

KONSTANTIOS P. FERENTINOS. “Deep learning models for plant disease detection and diagnosis”, **Computers and Electronics in Agriculture**,Volume 145,2018,Pages 311-318,ISSN 0168-1699.

MA, J., DU,K., ZHENG, F., ZHANG, L., GONG, Z., & SUN, Z.(2018). “A recognition method for cucumber diseases using leaf symptom images based on deep convolutional neural network”, **Comput. Electron. Agricult.**, vol. 154, pp. 18–24, Nov. 2018.

MERLINA, N., NOERSASONGKO, E., NURTANTIO, P., SOELMAN, M.A., RIANA, D., & HADIANTI,S. (2021). “Detecting the Width of Pap Smear Cytoplasm Image Based on GLCM Feature”, **Smart Innovation, Systems and Technologies. Vol. 182**. https://doi.org/10.1007/978-981-15-5224-3_22

MOHANTY, S. P., HUGHES D. P., SALATHE, M. “Using Deep Learning for Image-Based Plant Disease Detection”, **Journal - Frontiers in plant science**, vol-7,2016. https://www.frontiersin.org/articles/10.3389/fpls.2016.01419

MONIRIZZAMAN, M., ISLAM, S., LAVEY, P., BENNAMOUN, M. (2019).

"Faster R-CNN Based Deep Learnng for Seagrass Detection from Underwater Digital Images", **Digital Image Compuing (DICTA),** pages 1-7. 10.1109/DICTA47822.2019.8946048.

OUHAMI, M., HAFIANE, A., ES-SAADY, Y., & Canals, R. (2021). "Computer Vision, iot and Data Fusion for Crop Disease Detection Using Machine Learning: **A Survey and Ongoing Research." Remote Sensing** (13). https://doi.org/10.3390/rs13132486

OWOMUGISHA, G., QUINN, J., MWEBAZE, E., & LWASA. "Automated Vision-Based Diagnosis of Banana Bacterial Wilt Disease and Black Sigatoka Disease", **Preceding of the 1'st international conference on the use of mobile ICT in Africa ,2014.**

RAZA, S.E.A., SANCHEZ, V., PRINCE, G., CLARKSON, J., RAJPOOT, N.M. (2015). "Registration of Thermal and Visible Light Images of Diseased Plants Using Silhouette Extraction in the Wavelet Domain", **Pattern Recognition, 2015**. Volume 48, issue 7.

ROTHE, P. R. & R. V. KSHIRSAGAR. "Cotton Leaf Disease Identification using Pattern Recognition Techniques", **International Conference on Pervasive Computing (ICPC),2015.**

SANAKKI, S. S. and V. S. RAJPUROHIT. "Classification of Pomegranate Diseases Based on Back Propagation Neural Network", **International Research Journal of Engineering and Technology (IRJET)**, Vol2 Issue: 02 ,May-2015.

SIMONYAN, K., ZISSERMAN, A. (2015). "Very Deep Convolutional Networks for Large-scale Image Recognition", **Arxiv** . https://doi.org/10.48550/arXiv.1409.1556

SLADOJEVIC, S., ARSENOVIC, ANDERLA, A.,CULIBRK, D., STEFANOVC,D. "'Deep neural networks based recognition of plant diseases by leaf image classification", **Comput. Intell. Neurosci., 2016.** Article 3289801

SZEGEDY, C., VANHOUCKE, V., LOFFE, S., SHLENS, J., WOJNA, Z. "Rethinking the Inception Architecture for Computer Vision", **Arxiv.** https://doi.org/10.48550/arXiv.1512.00567

TAN, M., QUOC, V.L. (2019). “EfficientNet: Rethinking model scaling for convolutional neural networks”, **International Conference on machine learning.** https://doi.org/10.48550/arXiv.1905.11946

TIAN, U., ZHAO, C., LU, S., & GUO,X. “SVM-based Multiple Classifier System for Recognition of Wheat Leaf Diseases”, **Proceedings of 2010 Conference on Dependable Computing (CDC’2010), November 20-22, 2010.**

TURKOGLU, M., YANIKOĞLU, B., & HANBAY, D.(2022). “PlantDiseaseNet: convolutional neural network ensemble for plant disease and pest detection”, ***SIViP*** **16**, 301–309 ,2022. https://doi.org/10.1007/s11760-021-01909-2

ELECTRONIC SOURCES

BHIKSHA, R., “Courses”, https://www.cs.cmu.edu/~bhiksha/courses/10-601/decisiontrees/

BRONA. (2007). “Entropy plot”, https://en.wikipedia.org/wiki/Entropy_%28information_theory%29#/media/File:Binary_entropy_plot.svg

GHANTIWALA, A. (2022). “TowardsScience”, https://towardsdatascience.com/visualizing-audio-data-and-performing-feature-extraction-e1a489046000

MBAABU, O. (2020). “Introduction to machine learning”, https://www.section.io/engineering-education/introduction-to-random-forest-in-machine-learning/

ROHIT, G. (2018). “Support Vector Machine”, https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47

SIDDARTH, S., NAMAN, S. (2022). “Machine Learning”, https://www.researchgate.net/publication/358306897_Using_Deep_Learning_to_Bootstrap_Abstractions_for_Hierarchical_Robot_Planning

RESUME

Name Surname: Muhammad Umar MURAD

EDUCATION:

- **Bachelor**: 2020, Lahore Leads University, Faculty of Computer Sciences, Department of Software Engineering
- **M.A**: 2023, Istanbul Aydin University, Faculty of Computer Sciences, Department of Software Engineering, Artificial Intelligence and data science (English)(Thesis) Program

PROFESSIONAL EXPERIENCE:

- **Sales Specialist**
  Artin Marketing | Istanbul, Türkiye Dec 2020 – Present
- **Graphic Designer/ Cinematographer**
  Muslim Hands UK| Istanbul, Türkiye Sep 2020 – Dec, 2020
- **Graphic Designer**
  Roshan Studio| Okara, Pakistan Jan 2020 – July, 2020
- **Web Developer**
  MCB BANK LTD| Lahore, Pakistan Jun 2019 – Jan, 2020

AWARDS:

- **Home Automation System (IOT)**
  Lahore Leads University 23 Feb, 2015

- **Essay Writing Competition (English)**

- Govt. of Punjab, Pakistan 03 April, 2013

- **Essay Writing Competition (Urdu)**
  Govt. of Punjab, Pakistan 09 March, 2012

PUBLICATIONS FROM DISSERTATION, PRESENTATIONS AND PATENTS:

- MUHAMMAD UMAR MURAD, PROF. DR. ALI OKATAN (2023).
  Smart Detection and Diagnosis of Plant Disease Using Deep and Machine Learning Methods. **International Research Journal of Modernization in Engineering Technology and Science. Volume:05/Issue:01/January-2023**. https://www.doi.org/10.56726/IRJMETS33187