Multi-Layer Convolutional Neural Network Based Identification of Plant Category and Leaf Diseases

Author

HAMMAS BIN NAEEM

Regn Number

00000203565

Supervisor

Dr. M. Jawad Khan

ROBOTICS AND INTELLIGENT MACHINE ENGINEERING

SCHOOL OF MECHANICAL & MANUFACTURING ENGINEERING

NATIONAL UNIVERSITY OF SCIENCES AND TECHNOLOGY

ISLAMABAD

FEBRUARY 2021

Multi-Layer Convolutional Neural Network Based Identification of Plant Category and Leaf Diseases

Author

HAMMAS BIN NAEEM

Regn Number

00000203565

A thesis submitted in partial fulfillment of the requirements for the degree of

MS Robotics & Intelligent Machine Engineering

Thesis Supervisor:

Dr. M. Jawad Khan

Thesis Supervisor's Signature: ____

ROBOTICS AND INTELLIGENT MACHINE ENGINEERING

SCHOOL OF MECHANICAL & MANUFACTURING ENGINEERING

NATIONAL UNIVERSITY OF SCIENCES AND TECHNOLOGY

ISLAMABAD

FEBRUARY 2021

Thesis Acceptance Certificate

It is certified that the final copy of MS Thesis written by Hammas Bin Naeem (Registration No. 00000203565), of Department of Robotics and Intelligent Machine Engineering (SMME) has been vetted by undersigned, found complete in all respects as per NUST statutes / regulations, is free from plagiarism, errors and mistakes and is accepted as a partial fulfilment for award of MS Degree. It is further certified that necessary amendments as pointed out by GEC members of the scholar have also been incorporated in this dissertation.

Signature:____________________

Date:____________________

Dr. Muhammad Jawad Khan (Supervisor)

Signature HOD:____________________

Date:____________________

Signature Principal:____________________

Date:____________________

FORM TH-4

National University of Sciences & Technology

MASTER THESIS WORK

We hereby recommend that the dissertation prepared under our supervision by:

Hammas Bin Naeem, NUST Regn # 00000203565, Titled: **"**Multi-Layer Convolutional Neural Network Based Identification of Plant Category and Leaf Diseases**"** be accepted in partial fulfillment of the requirements for the award of **MS Robotics and Intelligent Machine Engineering** degree in _____ Grade.

Examination Committee Members

1. Name: Dr. Yasar Ayaz Signature:________________

2. Name: Dr. Hasan Sajid Signature:________________

3. Name: Dr. Umar Ansari Signature:________________

Supervisor's Name: Dr. Muhammad Jawad Khan Signature: ________________

Head of Department

Date

COUNTERSIGNED

Date: ___________

Principal

Declaration

I certify that this research work titled "*Multi-Layer Convolutional Neural Network Based Identification of Plant Category and Leaf Diseases"* is my own work. The work has not been presented elsewhere for assessment. The material that has been used from other sources it has been properly acknowledged / referred.

Signature of Student

HAMMAS BIN NAEEM

00000203565

Plagiarism Certificate (Turnitin Report)

This thesis has been checked for Plagiarism. Turnitin report endorsed by Supervisor is attached.

Signature of Student

HAMMAS BIN NAEEM

00000203565

Signature of Supervisor

Acknowledgements

First of all I am truly thankful to Allah the Almighty for all the blessings throughout my life especially in my educational career. Allah has always guided me towards the path which was best for me in all aspects. Pursuing my master's degree and completing the course work, research work would not be possible without the guidance of Almighty Allah.

I would love to express my heartily thanks to my Thesis Supervisor "Dr. M. Jawad Khan" for trusting in me, by providing me the opportunity to work with him. His guidance, support and availability at every part of my research phase made me complete this work. I would like to thank him for guiding me in all the events other than my research work which has helped me in grooming my technical skills and being motivated. His efforts and dedication throughout my research work were the key element for this accomplishment.

I would like to acknowledge my committee members Dr. Hasan Sajid, Dr. Omer Ansari and Dr. Yasar Ayaz for their support and encouragement throughout my course work and research work.

I am very thankful to my parents for supporting and loving me throughout my life by the best they could do for me. Specially my father Dr. Naeem Ullah Dar who has always motivated me by providing a practical example of hard work, honesty, dedication and commitment towards every phase of life.

I would also like to thank my friends for supporting and motivating me especially Mr. Zaid Ismail for his availability and help throughout my Master’s degree.

Dedicated to my family, specially my parents for their endless love and support throughout my career.

Abstract

Agriculture plays a major role in developing and improving the economy of a nation. It is the backbone of the global economy, the world's largest food source, and is responsible for ensuring a decent income for millions of homes worldwide. Pakistan is one of those countries in which agriculture plays a vital role in the development by assisting 19 percent of the Gross Domestic Product (GDP) and 42 percent of the labor employment approximately. As the world is going through major technological reforms in all its sectors including agricultural. Artificial Intelligence (AI) is one of the leading technologies which has been adopted by the most of the world in various domains. Machine Learning (ML) and Computer Vision (CV) has eased up the process of visualizing all types of data and providing the best outcomes from it. Agricultural sector growth is effected by various factors. Plant diseases are one of the leading factors. Plant diseases reduce crop yield and reduce production quality. There are various leaf diseases which cannot be identified through naked eye and it is a very challenging task for the farmers to keep information of all these diseases eventually leading to reduction of quality and overall production. Various research has been made in the field of agriculture using CV, ML and Deep Learning (DL) which includes crops disease detection, plant category identification, Leaf Disease Detection etc. No such study has been made on the combined identification of plant category and their leaf diseases. In this study we present a DL method by using a Multi-Layer Convolutional Neural Network for the identification of plants leaf diseases and their categories. Convolutional Neural Network (CNN) is a method which is being widely used for the classification of images and produces best results for many classification problems. Dataset was collected by two online available resources, Plant

Village and Fruits 360. Both the datasets were combined by taking the common plants classes including different images of Fruits, Vegetables and their Leaf diseases eventually leading to a final dataset of 70 classes having 167k images approximately. Dataset has been preprocessed according to the requirements of the proposed Convolutional Neural Network. A novel CNN has been proposed in this study for the classification of the acquired dataset. Several CNN configurations have been used for training, validating and testing the data. We have achieved an overall training and validation accuracies of 99.95 and 99.53 percent respectively. Our model is also tested on a batch of test images providing the best test accuracy of 99 percent.

Keywords: *Gross Domestic Product, Artificial Intelligence, Computer Vision, Machine Learning, Deep Learning, Convolutional Neural Network*

Table of Contents

List of Figures

List of Tables

List of Equations

List of Acronyms

CNN Convolutional Neural Network	CNN
Rectified Linear Unit	Relu
FC Fully Connected	FC
Stochastic Gradient Descent	SGD
Nestrovs Accelerated gradient	NAG
Gross Domestic Product	GDP
Food and Agricultural organization	FAO
Artificial Neural Network	ANN
Batch Size	BS
Data Split	DS
Machine Learning	ML
Computer Vision	CV
Artificial Intelligence	AI
Deep Learning	DL

1 Introduction

Agriculture assumes a major role in the development and improvement of a country's economy. It is the backbone of the world economy, the biggest provision of food and is responsible for providing millions of homes across the globe a sustainable income. According to analyses conducted in 2016, sixty five percent of poor working adults living is based on agriculture sector [1]. The area represents a comparatively smaller portion of the world economy. In a 2012 report by World Bank agriculture represented a mere 2.8 percent of the global economy. Starkly, agriculture represents 19 percent (1.3 billion) of the human capital [2]. Interestingly, in 2014 agriculture had a whopping one-third share of the global gross-domestic product (GDP) [3]. Agriculture has the potential to increase incomes among the poorest up to four times compared to other sectors. This is because agriculture plays a bigger role in low- and middle-income countries. For example, in India, agriculture embodies eighteen percent of the overall national income and fifty four percent of the employment [4]. Over the past half century, agriculture has shifted from being major production of the high-income countries, using superior technologies, to today, the bigger chunk being produced by low- and medium-income countries of the Asia and Pacific region. For example, in 1961 countries like the US produced approximately forty three percent of the total agricultural sector output combined, to 24.6 percent of the global share in 2011. The Asia and Pacific region increased its share of 23.9 percent in 1961 to 44.7 percent in 2011 [2]. A couple reasons could be the rapid urbanisation and industrialisation in developed countries and the rising pressure on developing countries to cater to their fast growing populations [5].

40 percent of the world's land area is utilised for agricultural needs and in 2009-2011 only 0.78 percent of the world's cropland comes from the 100 least contributing countries. India, the United States, the Russian Federation, China and Brazil account for 42.1 percent of the total cropland; almost three quarters of the total world output coming from the top 20 countries with China taking the crown in almost every major crop according to (FAO) at 23.0 percent of total agricultural output with the United States ranking second at an output of 10.1 percent of the total global produce [2] [6].

The global population is growing rapidly, estimating a remarkable 9.1 billion by 2050 (FAO), and requiring an additional 70 percent growth in food production [7]. Therefore, sustainable agricultural growth and its protection is of paramount importance for the food requirement of the coming decades and will play a pivotal role in the poverty alleviation of the global economy.

However, agriculture-driven growth is plagued by various factors. One of the leading factors are plant diseases. Plant diseases reduce crop yield and cuts quality of the output [8]. Approximately, 20-40 percent of the global agricultural yield is succumbed to pathogens, animals, and weeds [9]. This loss of yield in turn affects the fabrics of trade and its policies either directly or indirectly [10]. This suggests, crop protection is just as important as increasing crop productivity. The latter has been the focus of attention in the past few decades because protection of crops is complex and often expensive. Food safety is now recognised in not only the developed world, but in the developing too [11]. This is because assessment of crop losses has been undermined over the past half century leading to confusion.

Confusion leads to fear, which leads to wrong decisions eventually causing more harm than good [12]. The identification and diagnoses of crop diseases are at the behest of the naked-eye inspection of farmers. Modern technologies such as high-throughput serological techniques like *enzyme-linked immunosorbent assay* (ELISA) and molecular methods such as *polymerase chain reaction* (PCR) [13] are either predominantly unavailable in majority of the low- and middle-income countries of South Asia and the Pacific or too expensive. The rise of several crop related illnesses influences the efficiency in the agriculture segment. To adapt to this issue and to make the farmers mindful to prevent the development of diseases in crops and to execute viable administration, crop diseases identification and diagnosis executes its critical job. Moreover, the majority of the farmers, especially in developing countries, are ill-equipped and lack sufficient knowledge for the welfare of the crops, so visual inspection technologies such as the visible light imaging and chlorophyll florescence imaging are seldom available due to their requirement of uniformity in crop and expertise to carry out the tests [13]. Therefore, the system of detecting, identifying and diagnosing is not only complex, but can be expensive and time consuming [14] More often than not, farmers are unable to detect or correctly diagnose a part of their field in time and as a result, major patches of the crops are rendered useless. Primarily, two types of factors hinder the wellbeing of plants: *diseases* and *disorders*. Diseases are caused by external biotic attacks by the likes of bacteria or fungi, while disorders are caused by abiotic elements such as temperature, rainfall and soil health [15]. Fungal diseases such as *Anthracnose* for instance, are common in plant leaves and a responsible for a cut in the quality and also the yield of the crop [16]. Another example is the mosaic virus that causes the leaves to have a speckled appearance. These outbreaks have a potential to reach global epidemic proportions threatening not only the livelihood of the farmers and the associated supply chain but also the food security of millions [17]. This fall in yield holds formidable economic repercussions, both financially and food and nutrition security-wise. The key to avoid concessions in yield is to detect, diagnose and eradicate the pathogen infestation in its early stages. This will hinder the spread of the disease to other parts of the field and limit the loss of yield considerably, both quantitatively and qualitatively.

The advent of neural networks was essentially in 1951 when Marvin Minsky and Dean Edmonds built the first neural network machine, with the ability to learn, called the *SNARC*. In 1958, the first *artificial neural network* (ANN), developed by Frank Rosenblatt, was introduced by the name of *The Perceptron* which later inspired the *convolutional neural network*. The technology failed to pick up steam at the start due to various inherent problem with the design. The key problems were that it lacked good theory, neural networks relied heavily on a bulk of data, high computational power was required and the tendency of the trained models to over-fit. Therefore, *Support Vector Machines* (SVM), took over the scene in the late-90s after Corinna Cortes and Vladmir Vapnik published their work on SVMs.

With the advancement of personal computers, being able to handle large computations and better understanding of the CNNs, have made them a go-to method to analyse visual imagery. The popularity of CNNs has boomed over the past decade and still rising as a means of recognising and classifying images and videos, recommendation systems [18] image analysis etc. CNNs are a subset of deep learning which is made of fully connected multilayer perceptrons (A class of ANNs). The popularity can also be attributed to visual databases such as *ImageNet* and *Kaggle*, providing the necessary data for the CNN models to train, working as a catalyst to further propel the boom of Artificial Intelligence (AI) [19].

Much progress has been made since the advent of Computer Vision (CV), Machine Learning (ML) and AI. A lot of work is constantly being done in these three fields and it is growing rapidly. Development of automated models in all fields are on the rise, harnessing the power of these three areas of study combined. A significant portion of Deep Learning (DL) techniques (special class of ML algorithms with multiple layers) are being applied in the agricultural sector for crop and crop-disease identification and recognition. Similar to ANNs, DL's progress was stunted by the technology of its time (late 1990s). Insufficient processing power and lack of large data sets obstructed its way to penetrate the technological field. Enormous decrease in costs and the increased availability of reliable data have boosted the popularity of DL in recent years. Special data resources such as *Kaggle* and *ImageNet* have catalysed the progress of DL. Not only do these databases provide large data sets, but also hold annual competitions to encourage researchers and scientists. These competitions introduced more accurate neural networks such as *AlexNet*, developed by Alex Krizhevsky in 2012. AlexNet reduced the error rate to less than half of the previous year's winner.

Several complex and tedious problems have been solved using ANNs, CNNs and DL. Such as, detection and classification of leaves and leaf-diseases [20] [21]. Sharada P. Mohanty, David P. Hughes and Marcel Salathé in [21] used a deep CNN model for different plant disease detection based on a dataset of 54,306 images from the Plant Village database. Others such as J Amara et al [22] have used self-made datasets to train their model using CNNs to classify banana-leaf diseases. U. P. Singh, S. S. Chouhan, S. Jain and S. Jain [23] have used a hybrid data set using both, the Plant Village database and self-made real-time data sets of mango leaves infected by Anthracnose Disease with an accuracy of 97.13 percent. Similarly, Bin Liu, Yun Zhang, DongJian He and Yuxiang Li [24] have used a self-made data set of 13,689 images to identify apple leaf diseases based on Deep CNNs. In this work we propose to use Deep CNN to not only identify various leaf-diseases but also identify fruits and vegetables and some of their associated diseases.

Figure 1 AI in Agriculture [25]

1.1 Problem Statement

Agriculture plays a vital role in the country's economy, as the world is progressing with enormous technological advancements in all the domains including the agricultural sector. Various diseases in crops cannot be identified by naked eyes and it is so challenging for a human to keep the information of many insignificant diseases which eventually effects the crop yield. The problem Statement of this work is developing an effective Multi-Layer Convolutional Neural Network for the identification of various Fruits & Vegetables categories and their Leaf Diseases by combining the common categories of two publicly available Datasets (Fruit 360 and Plant Village) consisting of 70 classes having around 167k images for classification so it could help the agricultural sector.

1.2 Objective

The main objective of this work is to identify the plant category and their leaf diseases using Convolutional Neural Networks. The main purpose of this work is to benefit the agricultural sector by providing an intelligent solution for crop and their leaf diseases classification.

- Collection of data from various sources
- Data augmentation
- Architecture design
- Achieving 98% above accuracy

1.3 Areas of Application

Following are the major areas of application of this work

- Agricultural Sector
- Food Industry
- Super Marts (For Sorting and Quality check)

1.4 Thesis Overview

In this work, Section 2 briefly explains the previous work done by several researchers and comprises all the study of different theories for this proposed work. Section 3 contains the complete methodology and Implementation including Data Set, Data Pre Processing, Network Design, Network Implementation and complete work flow. Section 4 includes the complete results acquired after implementing the Convolutional Neural Network with several configurations. Section 5 consists of discussion of the complete work. Section 6 concludes the entire work. Section 7 describes all the possible future work which can be held in this domain.

2 Literature Review

2.1 Previous Work

In 2013 HP Mr. Hrishikesh P. Kanjalkar, Prof. S.S.Lokhande [26] proposed a system for automatic detection and identification of leaf diseases using ANN classification by analysing RGB pictures of the plants. It was a robust model with an accuracy of 70-83 percent on a data set acquired from the online National Integrated Pest Management (IPM) database; breaking down each image into: 1. Colour Transformation, 2. Image Segmentation, 3. Feature Extraction and 4. Classification.

In 2015 Malvika Ranjan, Manasi Rajiv Weginwar, Neha Joshi, Prof. A.B. Ingole also used an ANN technique to detect and classify plant and leaf diseases on a self-made data set with an accuracy of 80 percent. Utilising hue, saturation and value (HSV) features by segmentation and then training an ANN on the distinguishable features. The working set was of healthy cotton leaves in combination with disease ridden cotton leaves using similar techniques as Kanjalkar et al.

U Mokhtar et al [20] in 2015 used a different approach. They used Support Vector Machines (SVM) to detect tomato leaves diseases. They used a self-made dataset and achieved an accuracy of 99.83 percent. They used a Gray-Level Co-occurrence Matrix (GLCM) for the detection and identification of the tomato leaf state, whether it is healthy or infected. The dataset consisted of 800 images altogether used for both training the model and testing. The performance evaluation was done by the N-fold cross-validation technique.

In 2016, SP Mohanty et al [21] used deep CNN infrastructures, AlexNet and GoogleNet, to detect plant diseases using a dataset from Plant Village of 54,306 images and 38 classes. They achieved an overall accuracy of 99.35 percent. Training the system for the identification fourteen well defined crop types and twenty six defined diseases.

Also in 2016, Srdjan Sladojevic, Marko Arsenovic, Andras Anderla, Dubravko Culibrk, and Darko Stefanovic [27] used deep CNNs to recognise plant diseases using leaf-image classification. The final model had the ability to successfully identify thirteen types of plant diseases a 96.3 percent of average accuracy. The dataset comprised of 30,880 images acquired from a range of sources via the internet and categorised into 13 classes altogether.

Mohammed Brahimi, Kamel Boukhalfa and Abdelouahab Moussaoui in [28] executed a study akin to [21], using AlexNet and GoogleNet to classify 14,828 images tomato plant from Plant Village, using 9 different classes to classify tomato diseases and symptom visualisation. They were able to achieve an accuracy of up to 99.185 percent.

Chad DeChant, Tyr Wiesner-Hanks, Siyuan Chen, Ethan L. Stewart, Jason Yosinski, Michael A. Gore, Rebecca J. Nelson, Hod Lipson [29] used multi CNNs (DL) on a self-made dataset to achieve an accuracy of 96.7 percent with a suggestion to deploy the system on an aerial- or ground-based vehicle to monitor the fields. They used the imagery of the dataset for the automation.

Yang Lu, Shujuan Yi , Nianyin Zeng , Yurong Liu , Yong Zhang [30] used Deep CNN for the identification of rice diseases using a self-made 500-image-dataset with 95.48 percent accuracy for the ten-fold cross-validation. They trained the CNN to identify 10 distinct rice diseases. In the future, they plan to apply other algorithms such as the restricted Boltzmann machine [31] for a better performing system.

Sue Han Lee, Chee Seng Chan, Simon Joseph Mayo, Paolo Remagnino [32] used the MK dataset to train their deep CNN AlexNet model in 2017. They trained the CNN to analyse how deep learning extracts and learns features of images to classify various plants with an accuracy of 96.3 percent. They do this by gaining intuition of chosen features that are based on a *deconvolutional Network* approach. They observed that different orders of venation represent the features better than an outline of the shape ad that multi-level representation in leaf data, suggesting a "hierarchal transformation of features".

In 2017, Amanda Ramcharan, Kelsee Baranowsk , Peter McCloskey, Babuali Ahmed, James Legg and David P. Hughes [33] employed deep learning (Inception V3) for identification of cassava diseases. Cassava is a source of carbohydrates vulnerable to diseases. They used a self-made dataset of a field in Tanzania to achieve an accuracy of up to 98 percent with an overall accuracy of 93 percent. They used the model to identify two diseases (cassava brown streak disease, cassava mosaic disease) and three types of pest damage.

Alvaro Fuentes, Sook Yoon, Sang Cheol Kim and Dong Sun Park [34] made a robust DL based detection of tomato plant disease in real time and pest recognition with a self-made dataset of 5000 images having various resolutions. They used three types of detectors, Faster R-CNN, R-FCN and SSD to find out which of the three is most suitable of the task. They used VGG net and ResNET (Residual Network) as deep feature extractors. They were able to effectively recognise 9 distinct types of diseases and pests with a mean AP of 83 percent, declaring Faster R-CNN accompanied by VGG-16 showing the best achievable results.

Guan Wang, Yu Sun, and Jianxin Wang [35] used VGG-16, VGG-19, Inception-v3 and ResNet50 on a Plant Village dataset of 50,000 images to automatically detect the dangerous effects of a plant disease. They were able to achieve an accuracy in the range of 80.0 - 90.4 percent with a spread of 38 class labels. The best performing algorithm was the fine-tuned VGG-16 model which achieved the upper accuracy of 90.4 percent.

Yu-Dong Zhang et al. [36] utilised a hybrid dataset consisting of both self-taken images and internet-acquired images (3,600 images total) to train a deep CNN. They categorised 18 types of fruits using a 14-layer deep CNN and data augmentation with an overall max-pooling accuracy of 94.94 percent. Data augmentation methods used were, Gamma correction, image rotation and noise injection.

In 2018 Aravind Krishnaswamy Rangarajan, Raja Purushothaman, Aniirudh Ramesh [37] trained the AlexNet and VGG16 DCNNs with a Plant Village dataset to classify tomato crop diseases. The number of images used to train the algorithms was 13,262 and the accuracies for the two algorithms were 97.49 and 97.29 percent for AlexNet and VGG16 respectively.

Israr Hussain, Qianhua He, Zhuliang Chen [38] used a self-made dataset to train a DCNN model in order to automate fruit recognition, specifically for commercial source trace system. The dataset was 44,406 images large with 15 different categories of fruits including sub-categories, such as having 6 different kinds of apple fruits. They were able to achieve an accuracy of 99 percent and claim to "effectively meet real world application requirements."

Xihai Zhang, Yue Qiao, Fanfeng Meng, Chengguo Fan and Mingming Zhang [39] in 2018 proposed a system to identify maize-leaf diseases using GoogleNet and Cifar-10. The dataset of 500 images was acquired from Plant Village and other resources over the internet. They trained 9 different kinds of maize leaf images and achieved an accuracy of 98.9 percent on the GoogleNet model and 98.9 percent on Cifar10. Belal A. M. Ashqar and Samy S. Abu-Naser [40] used a dataset of 9,000 images to train a DCNN to identify 6 diseases associated with tomato leaves. They achieved an accuracy of 99.84 percent.

Also in 2018 Jayme G.A. Barbedo [41] researched the factors that influence the usage of DL on disease recognition of plants. The database used contained almost 50,000 images. According to the study, shortcomings highlighted by Barbedo can be marginalised by increasing available data for training. It suggested to contribute further in order to create comprehensive datasets. Another suggestion was to include the facilitation of farmers and other experts related to the field to label data better.

Konstantinos P. Ferentinos [42] researched deep learning models for plant disease detection. Using 87,848 images of 25 different plants in a set of 58 distinct classes. The models used to train were a variety but the VGG CNN was the most successful with an accuracy of 99.53 percent. Further concluding that the low computational power required by the trained model makes it feasible to deploy on mobile devices.

Dor Oppenheim, Guy Shani, Orly Erlich and Leah Tsror [43] used deep learning to detect and identify potato tuber disease. The self-made dataset of 2,465 images was divided into four classes of diseases and a fifth class of healthy potatoes. They were able to achieve an accuracy of up to 96 percent on the faster R-CNN after fine-tweaking the algorithms and its required image-set.

Mónica Pineda, María L. Pérez-Bueno and Matilde Barón [44] were able to successfully detect bacterial infection in melon plants. They used classifying methods to classify imaging data acquired from a self-ade dataset of 2,465 images, dividing them into 5 distinct classes. They used three algorithms to classify the images, logistic regression analysis (LRA), support vector machine (SVM), and ANN. They compared the performance of the three algorithms using the same dataset. The three had varying performance depending on the dosage of the plants and the variation was between 96 and 99 percent.

Bin Liu, et al. [24] also conducted a research in 2018 to identify apple-leaf diseases based on DCNNs. Their DCNN was based on AlexNet using self-made dataset of 13,689 images with 5 different classes. The model was able to achieve an accuracy of 97.63 percent, stating that the novel image generation technique proposed in the paper can make the model more robust.

Edna Chebet Too, Li Yujiana, Sam Njukia and Liu Yingchun [45] published a comparison of using fine-tuned DL models for the identification of plant diseases. The architectures used for the comparison were VGG16, Inception V4, ResNet with varying layers and DenseNet with 121 layers with the employment of Keras with Theano to train the architectures. The dataset used was acquired from plant village consisting of 54,306 images divided into 38 classes. They concluded that the DenseNets performed the best and kept increasing its accuracy as the number of epochs grew, ultimately achieving an accuracy of 99.75 percent.

Shanwen Zhang, Wenzhun Huang and Chuanlei Zhang [46] designed a three-channel convolutional neural network (TCCNN) for vegetable leaf-disease recognition. This architecture makes use of the colour information as well. The method avoids complex pre-processing techniques and the high-level discriminant features are extracted automatically. The Plant Village dataset was 15,817 images large and the model held out an accuracy of 94.27 percent.

In 2018, Horea Murşean and Mihai Oltean [47] used a fresh source of plant dataset from Fruit 360. We too, in our research have used this resource as one of the sources of our dataset. Horea et al. used the dataset of 69,905 images to train a CNN in order to recognise fruits from images. They achieved an overall accuracy of 94.16 percent.

In 2019 Jayme Garcia Arnal Barbedo [48] used GoogleNet to identify plant diseases from individual lesions and spots. The architecture was trained using 46,409 self-acquired images. The architecture was able to achieve at least 75 percent accuracy even when 10 different diseases were introduced. The conclusion was, that with a dataset big enough, deep learning techniques will be very effective for plant disease detection and recognition.

Aydin Kayaa et al. [49] Analysed the transfer learning for deep neural network based plan classification model. The approach aims to compliment the pre-existing knowledge, which had previously been used in isolation, of different applications of DNNs and integrate them with newer technologies. 4 different datasets were used, all publicly available, for this study. First two datasets were Flavia and Swedish Leaf, third was UCI Leaf dataset, and fourthly, the inclusion of the Plant Village dataset. Combined, the dataset had 54,000+ images for the training of the different architectures. Altogether, 5 DNN were used for the plant classification. First was self-developed, second was a fine-tuning the CNN, third was cross-dataset fine tuning, fourth was the use of pre-trained models, and lastly, fifth approach is the combination of recurrent neural network and CNN algorithms (AlexNet and VGG16). The best performance came out of the AlexNet/VGG-16 models when trained on the Plant Village dataset; an accuracy of 99.80 percent.

Shanwen Zhanga, Subing Zhangb, Chuanlei Zhangc, Xianfeng Wanga and Yun Shia [50] in 2018 published an article for the identification of cucumber leaf disease with a global pooling dilated convolutional neural network (GPDCNN). The reason for the use of GPDCNN over other CNNs such as the Alex Net is that convolution receptive field is increased without increasing the computational complexity, spatial resolution is recovered without an increase in parameters and the integration of dilated convolution and global pooling. They used a self-made dataset consisting of 65,000 images, divided into 7 classes to achieve an accuracy of 94.65 percent.

Geetharamani G., Arun Pandian J. [51] reported a research into identifying plant leaf diseases using deep CNN having nine layers. The model is trained using a Plant Village dataset of 61,486 images divided into 39 classes. Using techniques such as data augmentation they were able to achieve an accuracy of 94.46 percent.

Peng Jiang, Yuehan Chen, Bin Liu, Dongjian He, And Chunquan Liang [52] proposed real-time detection of apple-leaf diseases using deep learning based on *improved* CNNs. They used a self-made dataset of 26,377 images split into 5 classes to train an SSD with Inception module and Rainbow concatenation (INAR-SSD). They were able to achieve an accuracy of 78.8 percent.

Also in 2019, Uday Pratap Singh et al. [23] Published an article suggesting the use of multilayer convolutional neural network (MCNN) for the classification of mango leaves infected by the Anthracnose disease. They used a hybrid dataset of both real-time imagery (1070 images) and 1130 images from the Plant Village dataset. They achieved an accuracy of 97.13 percent claiming to be better than state-of-the-art infrastructure. Future deployment and expansion f the project was also discussed.

Yosuke Toda and Fumio Okura [53] discussed how CNNs diagnose plant diseases in an attempt to demystify the *black box*. They broke down the layers of the NN and revealed that neural networks capture the colours and textures of lesions upon diagnosis – akin to human decision making. They exposed layer not contributing to the decision making and removed them. Eventually they were able to reduce 75 percent of network parameters without affecting the accuracy. They used the Inception V3 algorithm with a dataset of 54,306 images divided in 38 classes.

Md. Rasel Mia, Sujit Roy, Subrata Kumar Das and Md. Atikur Rahman [54] used both an SVM algorithm and a CNN for mango-leaf disease detection. The detected 4 types of diseases making it a total of 5 classes, including the healthy group.

A. Blessy and Dr. D.C. Joy Winnie Wise [55] used disease segmentation to isolate the diseases region of the plant leaves and then fed these results into the CNN to classify the type of disease, propagating the final result onto a GSM device. They were able to achieve an accuracy of 98 percent.

Sanjay Sunkad, Aakash Bagale, Srujan Rumale and Zabiulla Shariff [56] identified different types of leaves for farmers using a multiclass classification method based on deep learning. The architectures used were Inception V3 on a self-made dataset, achieving an accuracy of 93.76 percent.

Marko Arsenovic, Mirjana Karanovic, Srdjan Sladojevic, Andras Anderla and Darko Stefanovic [57] published an article in 2019 in an attempt to solve the present limitations of DL based methods for disease detection and identification. They used a hybrid dataset from sources such as plant disease and plant village dataset and self-augmented images using Pro GAN bring the total of the images used to 139,011. They split this data into 42 distinct classes of diseases and plants. Their real breakthrough, they claim, is the use of a new dataset that allows the CNN to successfully detect images in real-world scenarios and angles. The architecture used was AlexNet and Solo achieving an accuracy of 93.67 percent.

2.2 Convolutional Neural Networks

According to Li Deng and Dong Yu in [58], deep learning is a class of machine learning algorithms that uses raw input to extract features by utilising several layers. The features extracted are of higher level, incorporating greater detail into the model. Deep learning was introduced to machine learning by Rina Dechter in 1986 in [59]. Since then various developments have taken place overtime such as the daw of neural networks working in both supervised and unsupervised conditions. Deep learning or convolutional neural networks (CNN) are part of the unsupervised realm of machine learning.

Among the pioneers of the development were Y. Lecun, L. Bottou, Y. Bengio and P. Haffner who developed the LeNet [60]. It was a 7-level convolutional network used to classify handwritten digits on cheques. Its major constrain was the high computational requirement. Later in 2012 Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton developed the CNN, AlexNet [61] that won the annual ImageNet Large Scale Visual Recognition Challenge (ILSVRC) held by ImageNet. The network had an architecture akin to LeNet but was deeper and consisted of more filters. Filters included, 11x11, 5x5, 3x3, convolutions, max pooling, dropout, data augmentation, ReLU activations and SGD with momentum. Then in 2013, the

ILSVRC was won by ZFNet, also a CNN, developed by Zeiler, Matthew D., and Rob Fergus [62] achieving a top-5 error rate of 14.8 percent, better than last year's AlexNet which had a top-5 error rate of 15.3 percent. This was achieved by primarily fine-tuning and tweaking the hyper-parameters of the AlexNet architecture. In 2014, the competition was won by GoogleNet, codenamed in the journal as *Inception V1* [63]. It made the first big leap after the AlexNet in terms of a top-5 error rate of 6.67 percent. It is based on the LeNet architecture and used batch normalisation, image distortions ad RMSprop. This novelty is dubbed as the *Inception Module*. This worked on reducing the number of parameters, using a 22-layer deep CNN to reduce the parameter from 60 million of AlexNet to 4 million. The runner-up to the GoogleNet was the VGGNet developed by Simonyan, Karen and Zisserman, Andrew [64] of the Oxford Robotics Institute.

The VGGNet consisted of 16 convolutional layers of 3x3 convolutions with more filters than the AlexNet. Its uniform structure makes it a go-to for various applications as a baseline feature extractor. Then in 2015, ResNet took the the ILSVRC crown, formally called the Residual Neural Network (RNN) [65]. RNN introduces skip connections, also called gated units allows this neural network to uses 152 layers while retaining a computational complexity less than VGGNet. It achievd a top-5 error rate of 3.57 percent.

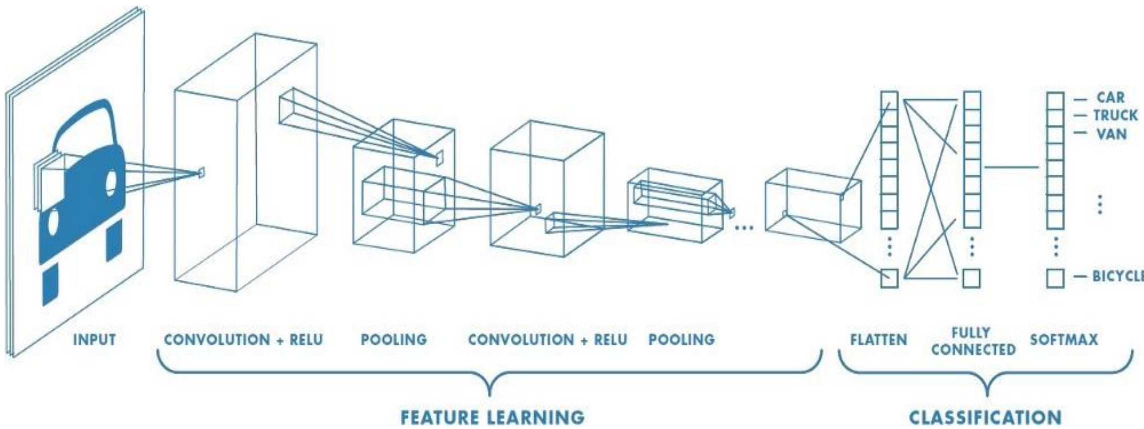

Figure 2 Convolutional Neural Network [66]

2.2.1 Convolution Layers

Convolutional Layers are responsible for the convolution of the Input Image and the filter to extract the required features and generates a feature map according to the filter size. Filter size is determined by the size of the Input Image. Filter consists of two parts the filter size *F* and the total amount of filters *K*. The input of the convolutional Layer would be the Input Image dimensions (*W(i) * H(i) * D(i)*) and the output (*W(o) * H(o) * D(o)*) where *D(o)* is equal to the total amount of filters *K* and *W(o)* and *H(o)* can be calculated by the following equation [67]

$$\frac{((W(i),\ H(i)) - F) + 2p}{Stride + 1}, \qquad (1)$$

Equation 1 Convolutional Layer Output Size

Where,

$W(i), H(i)$ is the Input Size of the square image

F is the Filter Size

p is the Padding

2.2.2 Parameter Calculation

Parameters for each convolution layer are calculated to get the overall trainable and non-trainable parameters in the model and to calculate the complete memory consumption of the network. If we have an input of ($W(i) * H(i) * D(i)$) and a convolution filter ($W(f) * H(f) * D(f)$) where $W(i)$, $H(i)$ and $D(i)$ are the Width, Height and Dimension of the input to the convolutional layer and $W(f)$, $H(f)$ and $D(f)$ are the width, height and total number of feature maps in a convolution filter. Thus the parameters can be calculated by using the following formula

$$(W(f) * H(f) * D(i) + 1) * D(f)\,, \qquad (2)$$

Equation 2 Parameter Calculation of Layers

2.2.3 Pooling Layers

Pooling Layers are used to reduce the total number of parameters which will be used further in the network and it also reduces the overall computational cost [67]. Most commonly used pooling techniques include Average Pooling and Max Pooling.

2.2.4 Dropout Layers

Dropout Layers were made to avoid overfitting or under fitting of the model on the given dataset. It chooses the amount of nodes which will be used in the training process. These Layers are commonly used after fully connected layers which are prone to overfitting [67].

2.2.5 Activation Function

Activation function helps in providing the non-linear relation between the class of image and Image Data. They determine that which neuron should be fired or not depends upon the relevancy of the neuron towards the required output [67]. Various activation functions are being used which includes tanh, sigmoid, Relu, Leaky Relu etc.

2.2.6 Optimization Techniques

Optimization techniques are used to calculate the weights for your model. They update the weights in the learning process until you reach towards your desired output. Various optimization techniques are used which includes SGD, SGD with momentum, NAG, Adagrad, RMSprop and Adam [67].

2.2.7 Flatten Layers

Flatten Layers are responsible for converting the data into one dimensional vector so it could be fed to Fully Connected Layers where classification will be completed.

2.2.8 Fully Connected Layers

These are the feed forward neural networks. The first FC layer collects the data from the last convolution Layer after getting flattened into one dimensional vector to compute the classification and the Last FC Layer provides the final probabilities calculated for each label.

2.2.9 Accuracy Calculation

It is calculated by using the f1 Score which has two metrics Precision and Recall [68]. Precision describes the number of true class predictions which truly belongs to the true class whereas recall defines the number of true class predictions completed out of all the true samples in the complete dataset.

Formulas for each is given below,

$$Precision = \frac{TP}{TP + FP}\,, \qquad (3)$$

Equation 3 Precision

$$Recall = \frac{TP}{TP + FN}\,, \qquad (4)$$

Equation 4 Recall

$$f1\ Score = 2\ .\frac{Precision\ .\ Recall}{Precission\ +\ Recall}\ , \qquad (5)$$

Equation 5 F1 Score

Where *TP* is True Positive, *FP* is False Positive, *FN* is False Negative

3 Methodology and Implementation

3.1 Proposed Scheme

In the proposed scheme we start with the collection of dataset from the available resources then this dataset be will processed according to the requirements which includes Image resizing, Data Augmentation, Combining the two different datasets and class labelling. After that data is to be distributed among Train and Validation sets and is fed to our proposed CNN. Later the trained model is tested on the individual images to verify the model performance.

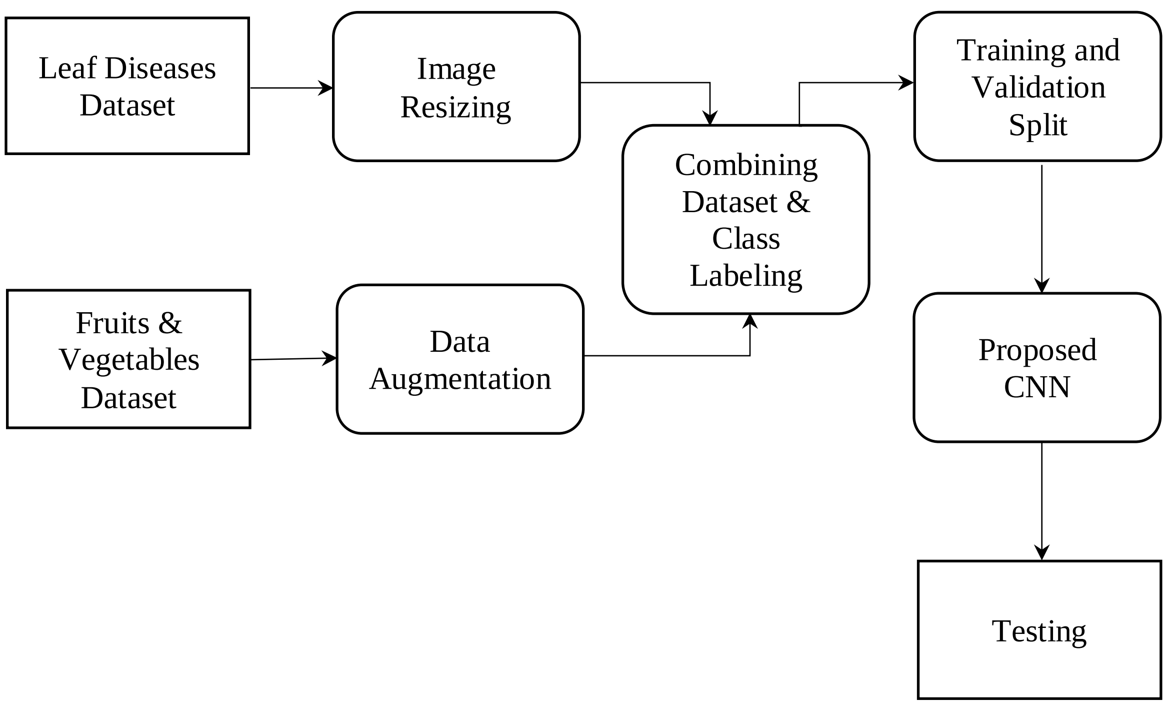

Figure 3 Proposed Scheme Diagram

3.2 Data Set

Dataset is obtained from two online publicly available resources Plant Village [69] and Fruit360 [47]. Plant Village dataset contains 38 classes of healthy and leaf diseases having 70,328 images for training and testing. Fruit360 dataset contains 114 classes of fruits and vegetables having 76,824 images for the classification. Some of the examples of plants and leaf diseases are given below

(a) (b) (c) (d) (e)

Figure 4 Plant Village Dataset [69]

(a) Apple Scab Leaf, (b) Cedar Apple Rust, (c) Cherry Powdery mildew, (d) Grape Black Measles, (e) Orange Haunglongbing (Citrus greening)

(a) (b) (c) (d) (e)

Figure 5 Fruit360 Dataset [47]

(a) Apple Breaburn, (b) Apple Golden 3, (c) Cherry 1, (d) Orange, (e) Peach

We have obtained the common classes from the both datasets which includes 30 classes from the Plant Village having 69,521 images and 40 classes from Fruit360 having 97,160 images for classification. So we get the overall of 70 classes including 166,681 for our classification problem. Class wise distribution of data is provided in the below Table

Sr. No	**Class Name**	**Number of Images**
1	Apple___Apple_scab_Leaf	2519
2	Apple___Black_rot_Leaf	2438
3	Apple___Cedar_apple_rust_Leaf	2200
4	Apple___healthy_Leaf	2509
5	Apple_Braeburn	2471
6	Apple_Crimson_Snow	2280
7	Apple_Golden_2	2474
8	Apple_Golden_3	2426
9	Apple_Granny_Smith	2470
10	Apple_Pink_Lady	2326
11	Apple_Red_1	2471
12	Apple_Red_2	2472
13	Apple_Red_3	2220
14	Apple_Red_Delicious	2464
15	Apple_Red_Yellow_1	2472

16	Cherry_(including_sour)___healthy_Leaf	2282
17	Cherry_(including_sour)___Powdery_mildew_Leaf	2104
18	Cherry_1	2474
19	Cherry_2	2475
20	Cherry_Rainier	2496
21	Cherry_Wax_Black	2471
22	Cherry_Wax_Red	2472
23	Cherry_Wax_Yellow	2474
24	Grape___Black_rot_Leaf	2359
25	Grape___Esca_(Black_Measles)_Leaf	2400
26	Grape___healthy_Leaf	2115
27	Grape___Leaf_blight_(Isariopsis_Leaf_Spot)_Leaf	2152
28	Grape_Blue	2498
29	Grape_Pink	2471
30	Grape_White	2467
31	Grape_White_2	2467
32	Grape_White_3	2472
33	Grape_White_4	2388
34	Orange	2421
35	Orange___Haunglongbing_(Citrus_greening)_Leaf	2513
36	Peach	2471
37	Peach___Bacterial_spot_Leaf	2296
38	Peach___healthy_Leaf	2160

39	Peach_2	2497
40	Pepper_bell___Bacterial_spot_Leaf	2391
41	Pepper_bell___healthy_Leaf	2485
42	Pepper_Green	2280
43	Pepper_Red	2504
44	Pepper_Yellow	2501
45	Potato___Early_blight_Leaf	2423
46	Potato___healthy_Leaf	2280
47	Potato___Late_blight_Leaf	2424
48	Potato_Red_Washed	2316
49	Potato_White	2304
50	Strawberry	2472
51	Strawberry___healthy_Leaf	2280
52	Strawberry___Leaf_scorch_Leaf	2218
53	Strawberry_Wedge	2500
54	Tomato___Bacterial_spot_Leaf	2127
55	Tomato___Early_blight_Leaf	2400
56	Tomato___healthy_Leaf	2407
57	Tomato___Late_blight_Leaf	2313
58	Tomato___Leaf_Mold_Leaf	2352
59	Tomato___Septoria_leaf_spot_Leaf	2180
60	Tomato___Spider_mites_Two-spotted_spider_mite_Leaf	2176
61	Tomato___Target_Spot_Leaf	2284

62	Tomato___Tomato_mosaic_virus_Leaf	2238
63	Tomato___Tomato_Yellow_Leaf_Curl_Virus_Leaf	2451
64	Tomato_1	2503
65	Tomato_2	2504
66	Tomato_3	2504
67	Tomato_4	2422
68	Tomato_Cherry_Red	2474
69	Tomato_Maroon	1976
70	Tomato_Yellow	2340
	Total Classes = 70	**Total Images** = 166,681

Table 1 Individual Class Dataset

3.3 Dataset Distribution

Dataset is distributed into three parts Training, Validation and Testing. Training and Validation data is split by the ratio of (70, 30), (75, 25) and (80, 20) respectively. Some test images where separated before the data distribution for individual testing.

3.4 Data Augmentation and Resizing

Data augmentation has been used to balance the classes of two different datasets so we could overcome the overfitting and lighting issues. The 40 classes from the Fruit 360 dataset were augmented by the state of art image processing operations which includes Rotation of images, Vertical Flip, Horizontal Flip, Brightness, Enhancement, and Sharpness. Plant Village dataset was resized to 100*100 resolution so we could obtain same resolution for the complete dataset.

3.5 Proposed Architecture

The proposed Convolutional Neural Network has 5 convolutional layers with a filter size of 3*3 having feature maps of 64, 128 and 256 with a stride of one respectively. Smaller filter sizes are used as we have smaller sized images so we could get the best possible informational features. First convolutional layers is followed by Max Pooling layers having filter size of 2*2 with a stride of two, second convolutional layer is followed by another Max Pooling layer having filter size of 3*3 with a stride of 2 and the last convolutional layer is followed by a Max Pooling Layer having filter size of 3*3 with a stride of 2. Relu is used for the activation of neurons. Flatten Layer is responsible for the conversion of last feature map into one dimensional vector so it could be fed to the neural network. Four dense layers have been used as a feed forward neural network including 2048, 512 and 70 neuron cells. Dropout has been used to avoid overfitting problems.

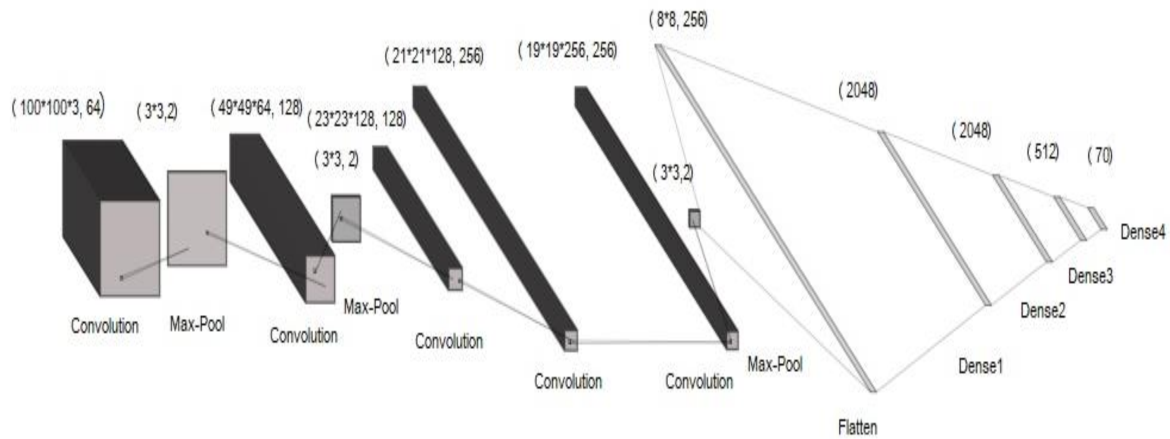

Figure 6 Proposed Convolutional Neural Network

3.5.1 Configuration Details of Proposed Architecture

The configuration details of the proposed architecture are provided in the following table

Convolutional Layers	5 (each with 3*3 filters & (64, 128, 256) feature maps
Max Pooling	3 (with 2*2 & 3*3 filters with stride of 2)
Dropout	0.5
Learning Rate	0.01
Training Algorithm	SGD
Momentum	0.9
Weight decay	1e-4
Activation Function	Relu
Batch Size	64, 128, 256

Dataset Distribution	(70, 30) (75, 25) (80, 20)
Number of epochs	35

Table 2 Configuration details of CNN

3.5.2 Architecture Summary

The overall inputs, outputs and the total number of parameters for each layer of the network are calculated by the above mentioned Equations (1) and (2). There are total of 39,968,070 number of parameters in which 39,957,190 are trainable and 10,880 are non-trainable. The complete summary is provided in the following table

Layer (type)	Output Shape	Number of Parameters
Convlayer1	(None, 98, 98, 64)	1792
active1	(None, 98, 98, 64)	0
batchnorm1	(None, 98, 98, 64)	256
Maxpool1	(None, 49, 49, 64)	0
Convlayer2	(None, 47, 47, 128)	73856
active2	(None, 47, 47, 128)	0
batchnorm2	(None, 47, 47, 128)	512
Maxpool2	(None, 23, 23, 128)	0
Convlayer3	(None, 21, 21, 128)	147584
active3	(None, 21, 21, 128)	0
batchnorm3	(None, 21, 21, 128)	512

Convlayer4	(None, 19, 19, 256)	295168
active4	(None, 19, 19, 256)	0
batchnorm4	(None, 19, 19, 256)	1024
Convlayer5	(None, 17, 17, 256)	590080
active5	(None, 17, 17, 256)	0
batchnorm5	(None, 17, 17, 256)	1024
Maxpool3	(None, 8, 8, 256)	0
flatten1	(None, 16384)	0
dense1	(None, 2048)	33556480
active5	(None, 2048)	0
batchnorm6	(None, 2048)	8192
dropout1	(None, 2048)	0
dense2	(None, 2048)	4196352
active6	(None, 2048)	0
batchnorm7	(None, 2048)	8192
dropout2	(None, 2048)	0
dense3	(None, 512)	1049088
activation7	(None, 512)	0
batchnorm8	(None, 512)	2048
dropout3	(None, 512)	0
dense4	(None, 70)	35910
active8	(None, 70)	0

		Total params: 39,968,070 Trainable params: 39,957,190 Non-trainable params: 10,880

Table 3 Proposed Architecture Summary

4 Results and Discussion

4.1 Batch Size of 64

This section discuss the results of the trained network on the given dataset for the Batch Size (BS) of 64 with the data split of (70, 30), (75, 25) and (80, 20) respectively.

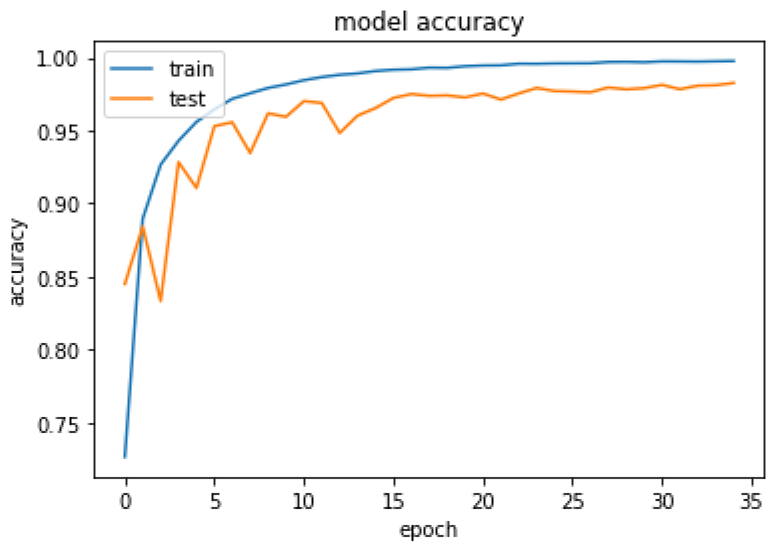

Figure 7 Model Accuracy for Data Split of (70, 30)

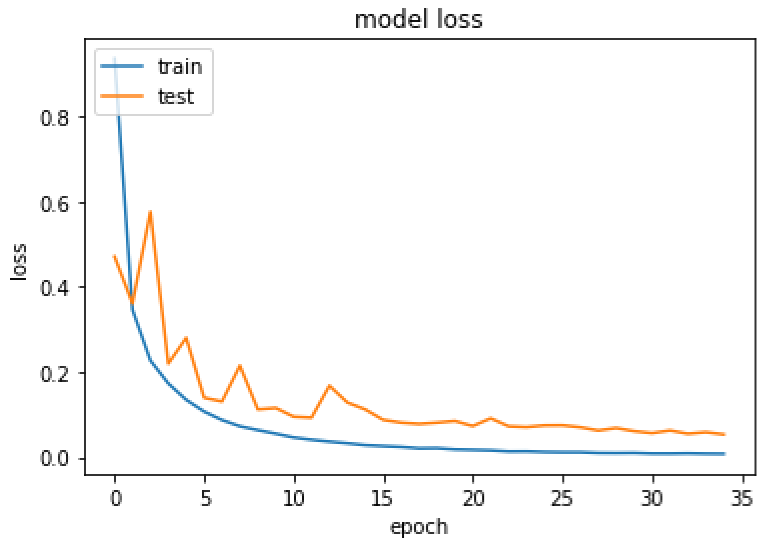

Figure 8 Model Loss for Data Split of (70, 30)

According to Figure (5) and (6) we can see that the training accuracy and validation accuracy for DS of (70, 30) of the model is 99.81 and 97.60 percent respectively. The model loss for training and validation is 0.64 and 3.15 percent.

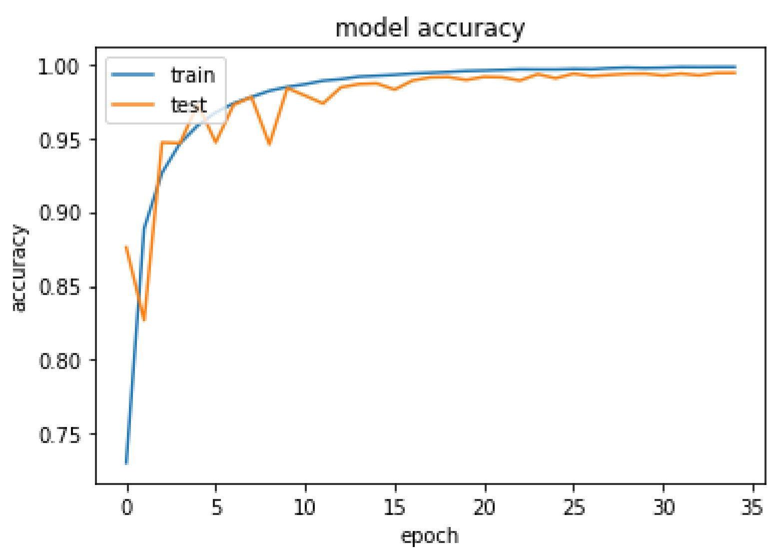

Figure 9 Figure 6 Model Accuracy for Data Split of (75, 25)

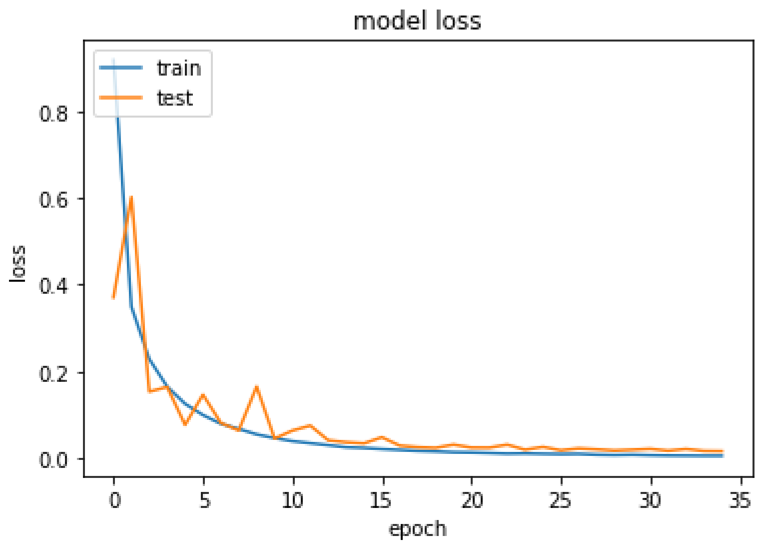

Figure 10 Figure 6 Model Loss for Data Split of (75, 25)

As we can see in the above Figure (7) and (8) the training accuracy and validation accuracy for DS of (75, 25) of the model is 99.89 and 99.48 percent respectively. The model loss for training and validation is 0.34 and 1.12 percent.

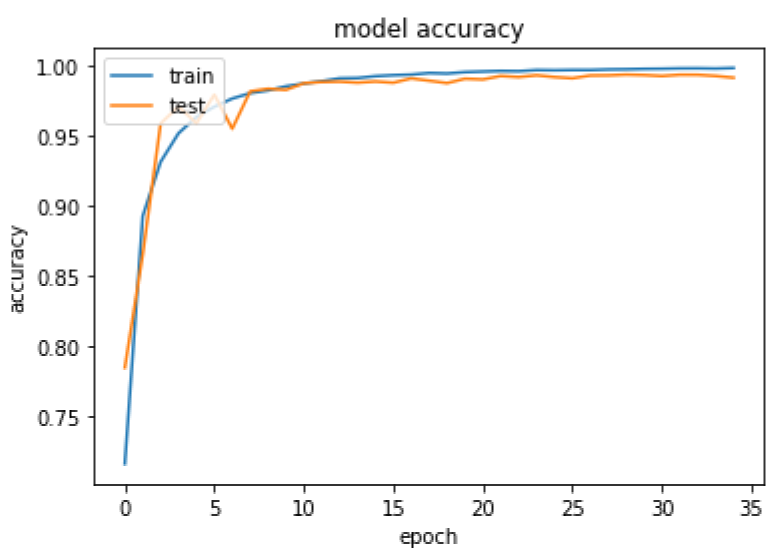

Figure 11 Model Accuracy for Data Split of (80, 20)

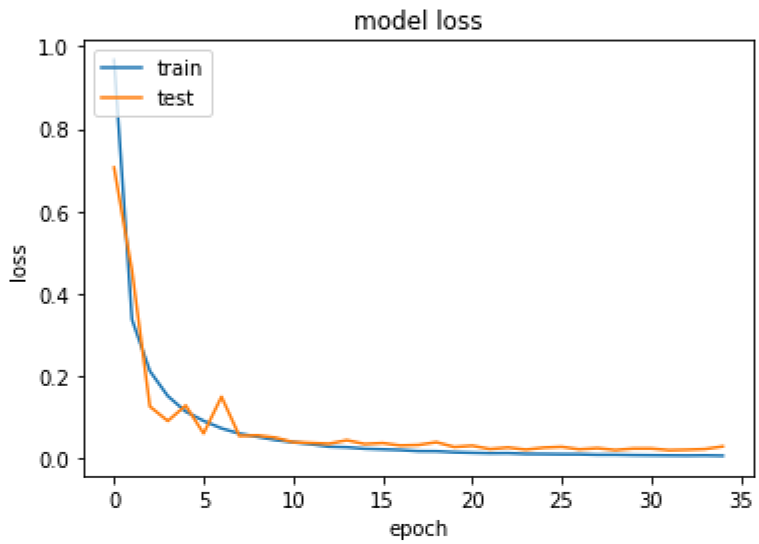

Figure 12 Model Loss for Data Split of (80, 20)

By the above Figure (9) and (10) we can conclude that the training accuracy and validation accuracy for DS of (80, 20) of the model is 99.95 and 99.53 percent respectively. The model loss for training and validation is 0.30 and 0.81 percent.

4.2 Batch Size of 128

The results of the trained network on the given dataset for the Batch Size (BS) of 128 with the data split (DS) of (70, 30), (75, 25) and (80, 20) are provided in this section.

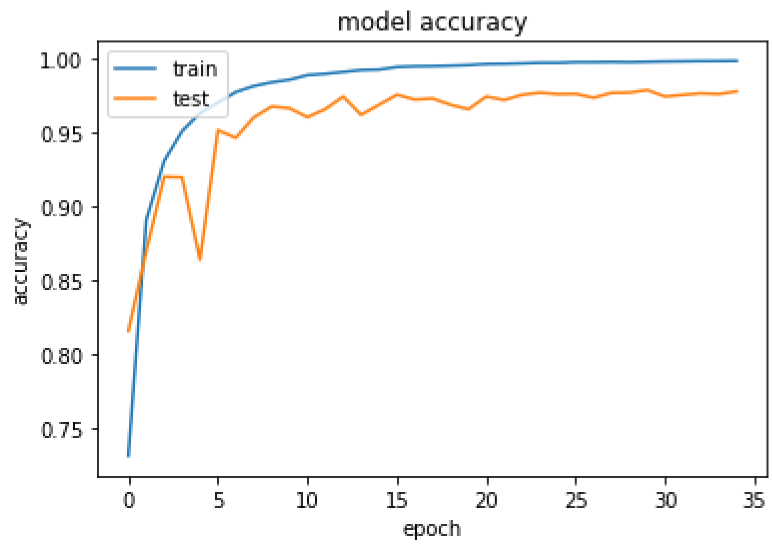

Figure 13 Model Accuracy for Data Split of (70, 30)

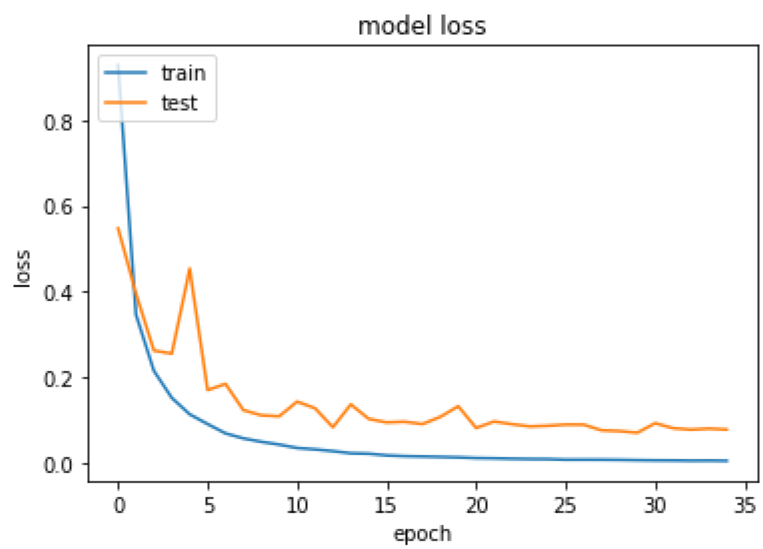

Figure 14 Model Loss for Data Split of (70, 30)

Figure (11) and (12) shows that the training and validation accuracy for DS of (70, 30) of the model is 99.30 and 96.80 percent respectively. The training and validation model loss is 0.74 and 6.32 percent.

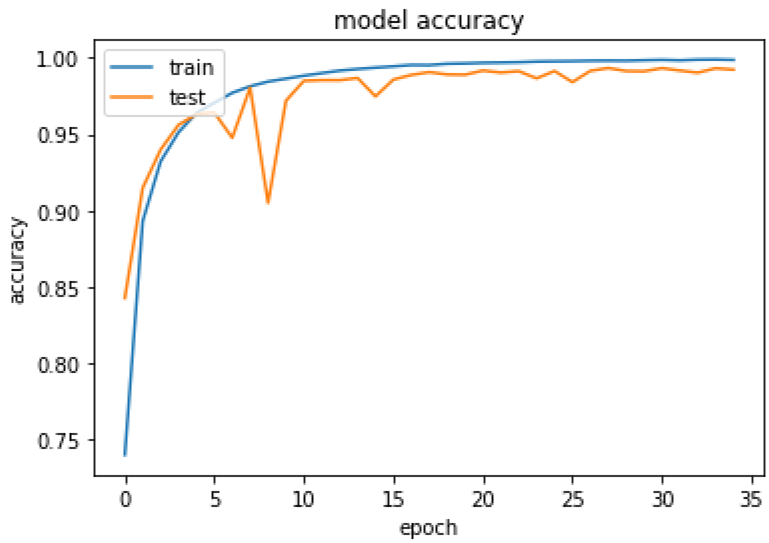

Figure 15 Model Accuracy for Data Split of (75, 25)

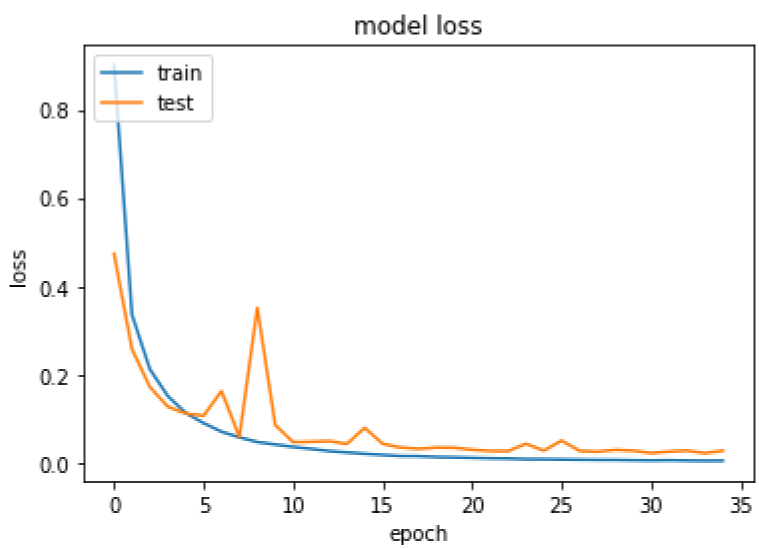

Figure 16 Model Loss for Data Split of (75, 25)

As we can see in the above given Figure (13) and (14) the training accuracy and validation accuracy for DS of (75, 25) of the model is 99.87 and 99.22 percent respectively. The loss for training and validation is 0.47 and 2.74 percent.

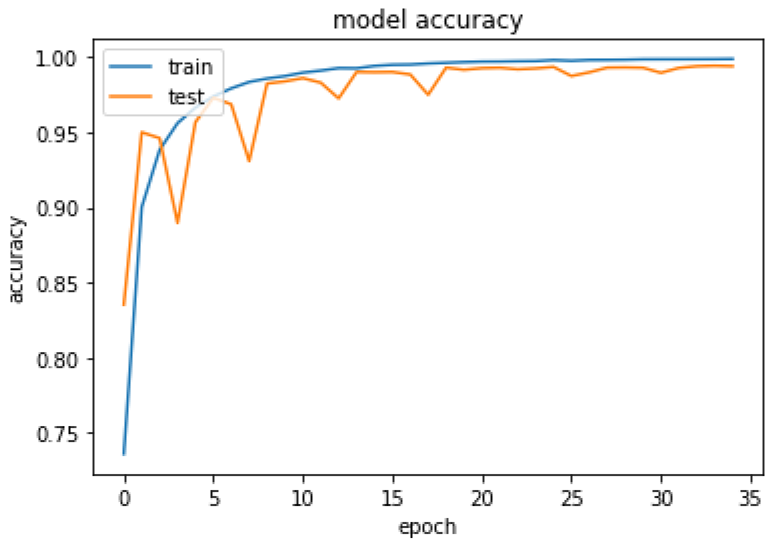

Figure 17 Model Accuracy for Data Split of (80, 20)

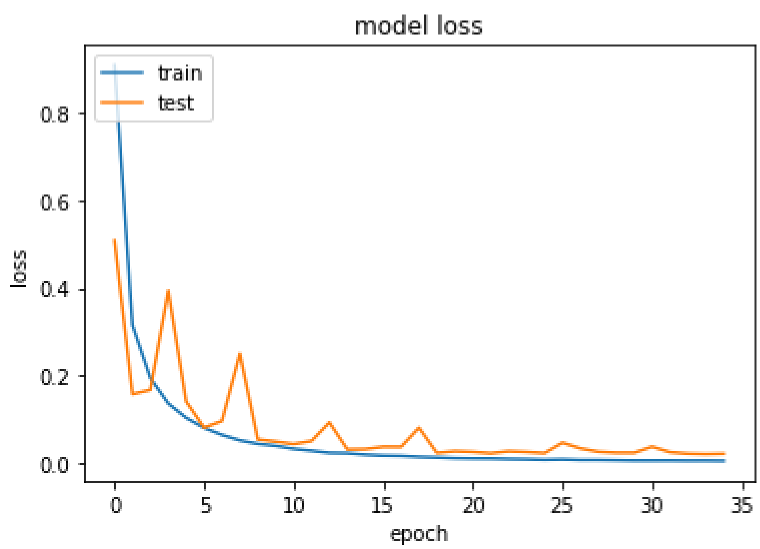

Figure 18 Model Loss for Data Split of (80, 20)

According to Figure (15) and (16) we can see that the training accuracy and validation accuracy for DS of (80, 20) of the model is 99.88 and 99.41 percent respectively. The model loss for training and validation is 0.43 and 2.05 percent.

4.3 Batch Size of 256

In this section the results of the trained network on the given dataset for the Batch Size (BS) of 128 with the data split (DS) of (70, 30), (75, 25) and (80, 20) respectively are discussed below

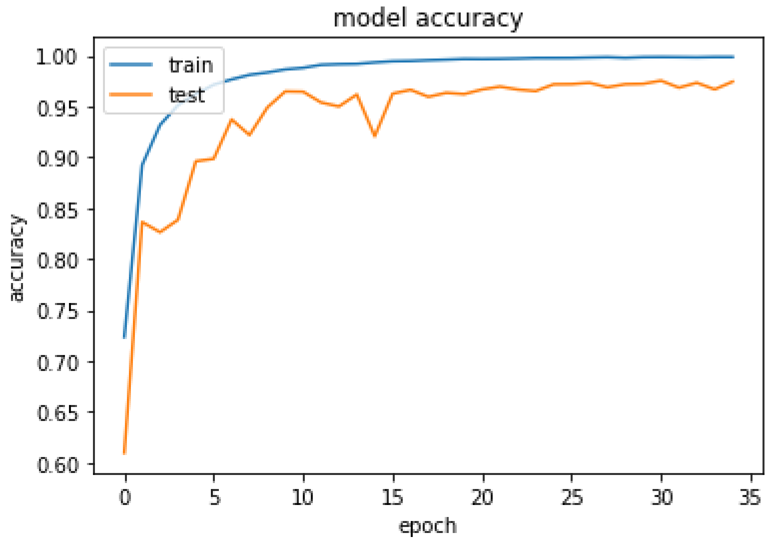

Figure 19 Model Accuracy for Data Split of (70, 30)

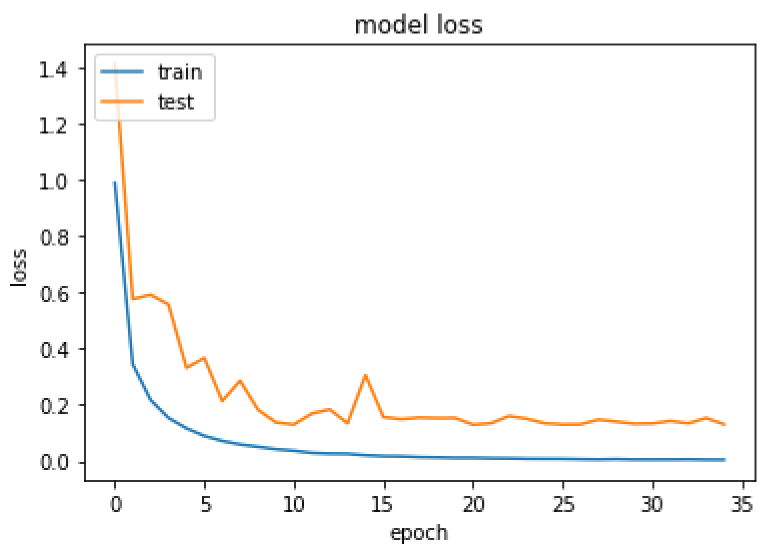

Figure 20 Model Loss for Data Split of (70, 30)

Figure (17) and (18) provides the information that the accuracies of training and validation for DS of (70, 30) of the model is 99.87 and 97.45 percent respectively. The model loss for training and validation is 0.46 and 13.07 percent.

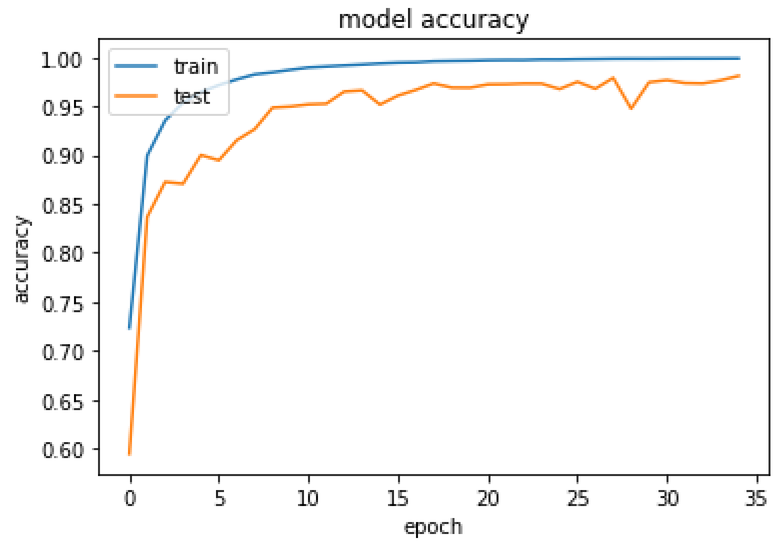

Figure 21 Model Accuracy for Data Split of (75, 25)

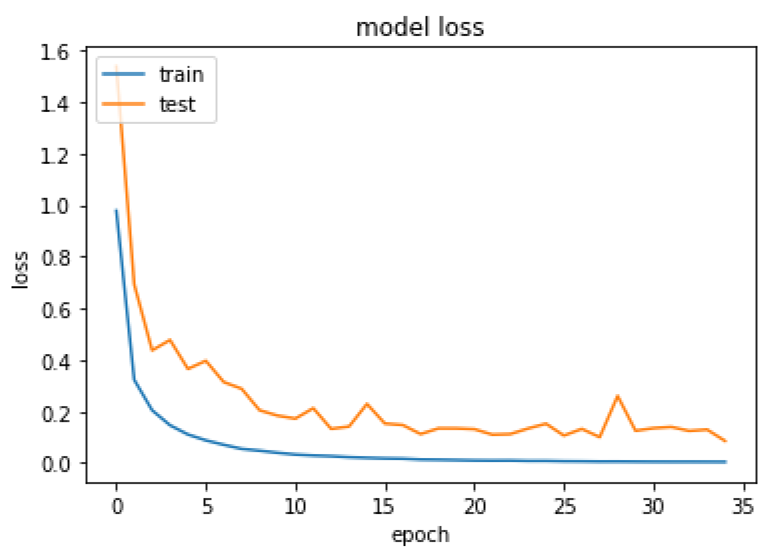

Figure 22 Model Loss for Data Split of (75, 25)

The above given Figure (19) and (20) we can say that the accuracies of training and validation for DS of (75, 25) of the model is 99.89 and 98.10 and the model loss for training and validation is 0.44 and 8.58 percent.

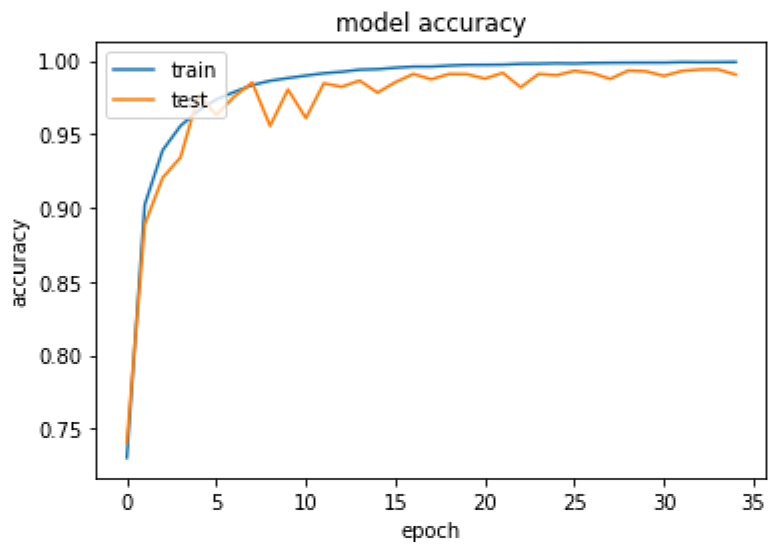

Figure 23 Model Accuracy for Data Split of (80, 20)

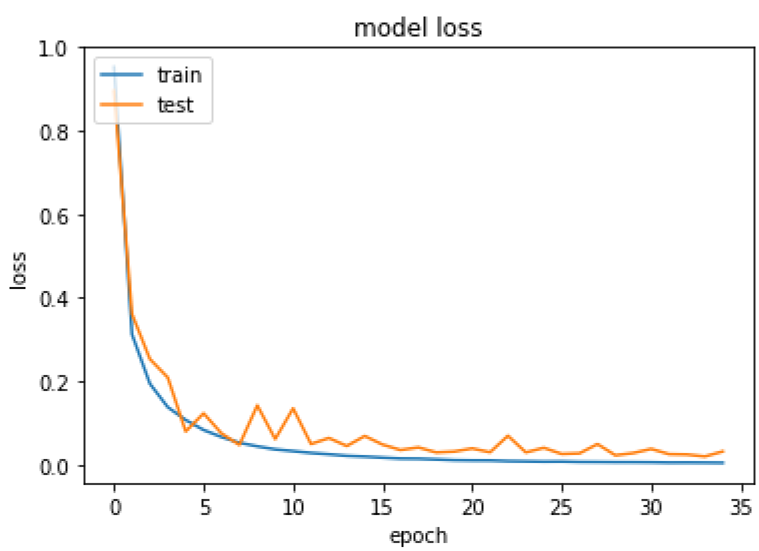

Figure 24 Model Loss for Data Split of (80, 20)

According to the information provided by the above Figure (21) and (22) we can see that accuracies of training and validation for DS of (80, 20) is 99.91 and 99.05 and the model loss for training and validation is 0.36 and 3.10 percent.

4.4 Comparison

According to the above mentioned results for different batch size (BS) and data split (DS) the comparison of overall performance in given below

BS (Batch Size)	DS (Data Split)	Training Accuracy (%)	Validation Accuracy (%)	Training Loss (%)	Validation Loss (%)
64	70, 30	99.81	97.60	0.64	3.15
	75, 25	99.89	99.48	0.34	1.12
	80, 20	99.95	99.53	0.30	0.81
128	70, 30	99.30	96.80	0.72	6.34
	75, 25	99.87	99.22	0.47	2.74
	80, 20	99.88	99.41	0.43	2.05
256	70, 30	99.87	97.45	0.46	13.07
	75, 25	99.89	98.10	0.44	8.58
	80, 20	99.91	99.05	0.36	3.10

Table 4 Results Comparison

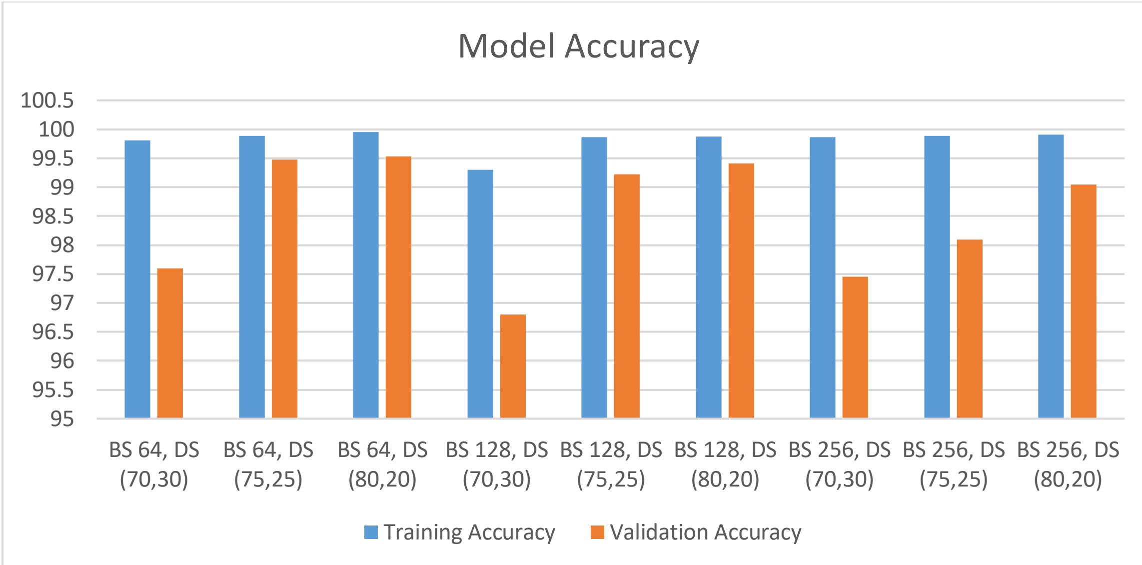

Figure 25 Accuracies Comparison

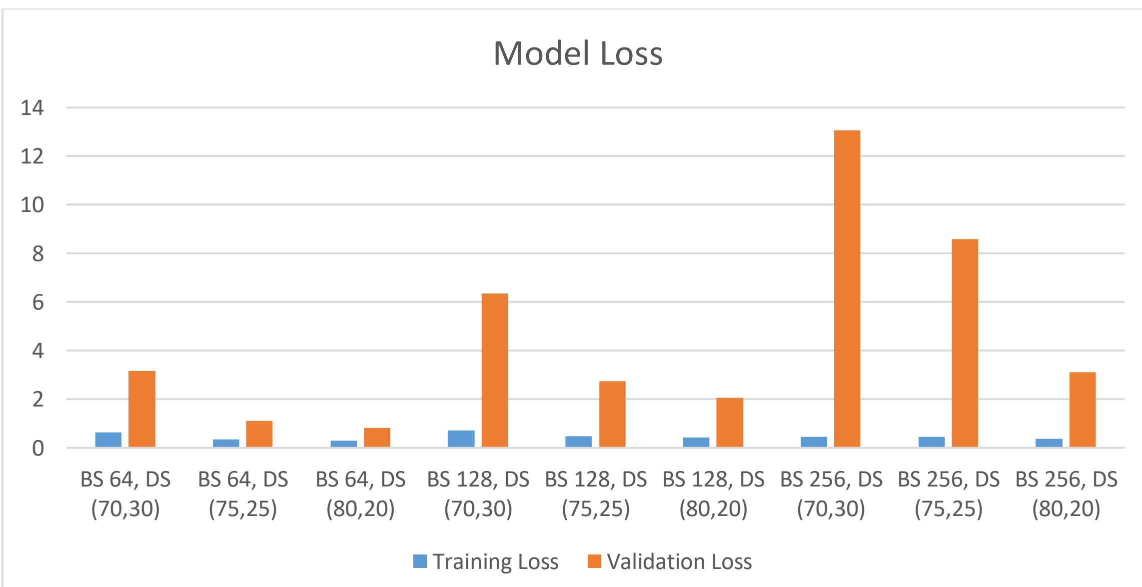

Figure 26 Loss Comparison

As we can see from the above given Table (5) and Figure (23) (24) the best performance our proposed model achieved is using the configuration of batch size (BS) and data split (DS) of 64 and (80, 20) respectively, by achieving the highest training and validation accuracies of 99.95 and 99.53 percent and by getting the lowest loss of 0.30 and 0.81 for both training and validation.

4.5 Classification Report

The model was also tested on a batch of 128 images and single images achieving the accuracy of 99 percent. The classification report for the model is given below

Class Number	Precision	Recall	F1-score	Support
1.0	1.00	1.00	1.00	1
2.0	0.67	1.00	0.80	2
3.0	1.00	1.00	1.00	3
4.0	1.00	1.00	1.00	4
5.0	1.00	1.00	1.00	1
6.0	1.00	1.00	1.00	2
7.0	1.00	1.00	1.00	1
8.0	1.00	1.00	1.00	2
9.0	1.00	1.00	1.00	4
10.0	1.00	0.67	0.80	3
11.0	1.00	1.00	1.00	5
12.0	1.00	1.00	1.00	4
14.0	1.00	1.00	1.00	1
15.0	1.00	1.00	1.00	2
17.0	1.00	1.00	1.00	1
18.0	1.00	1.00	1.00	1
19.0	1.00	1.00	1.00	3
20.0	1.00	1.00	1.00	2
21.0	1.00	1.00	1.00	2
22.0	1.00	1.00	1.00	1
23.0	1.00	1.00	1.00	2
24.0	1.00	1.00	1.00	2
25.0	1.00	1.00	1.00	1
26.0	1.00	1.00	1.00	3
27.0	1.00	1.00	1.00	1
28.0	1.00	1.00	1.00	3
29.0	1.00	1.00	1.00	1

32.0	1.00	1.00	1.00	1
33.0	1.00	1.00	1.00	1
34.0	1.00	1.00	1.00	2
35.0	1.00	1.00	1.00	3
36.0	1.00	1.00	1.00	6
37.0	1.00	1.00	1.00	2
38.0	1.00	1.00	1.00	2
41.0	1.00	1.00	1.00	1
42.0	1.00	1.00	1.00	1
43.0	1.00	1.00	1.00	3
44.0	1.00	1.00	1.00	1
47.0	1.00	1.00	1.00	4
48.0	1.00	1.00	1.00	2
49.0	1.00	1.00	1.00	1
50.0	1.00	1.00	1.00	2
51.0	1.00	1.00	1.00	5
52.0	1.00	1.00	1.00	2
53.0	1.00	1.00	1.00	1
54.0	1.00	1.00	1.00	1
55.0	1.00	1.00	1.00	2
56.0	1.00	1.00	1.00	3
57.0	1.00	1.00	1.00	6
58.0	1.00	1.00	1.00	1
59.0	1.00	1.00	1.00	1
61.0	1.00	1.00	1.00	2
62.0	1.00	1.00	1.00	2
63.0	1.00	1.00	1.00	3
64.0	1.00	1.00	1.00	1
65.0	1.00	1.00	1.00	5
66.0	1.00	1.00	1.00	1
67.0	1.00	1.00	1.00	1
68.0	1.00	1.00	1.00	1
Accuracy			0.99	128
Macro avg	0.99	0.99	0.99	128
Weighted avg	0.99	0.99	0.99	128

Table 5 Classification Report

We can see by the above Table (6) the macro and weighted averages are 99 percent.

Figure 27 Testing Result

Figure (27) represents some testing images which were classified true by giving the green classes names on the test images, if any of the image is classified wrong its true class name will be showed with red color.

5 Future Work

We can use the other Data Augmentation techniques and other new networks for this dataset to achieve lower loss. We can convert the data to higher resolutions so we can get more detailed results. We can use Faster RCNN and Mask RCNN for the detection of the infected area using bounding boxes.

We can add more data of infected Fruits/ Vegetables and the solution to the diseases e.g spray, so we could make a complete solution for the agriculture sector which will cover the following steps

- Identification of the Plant Category
- Identification of Plant Diseases
- Identification of Plant Leaf Disease
- Detection of infected area of the Plants and their Leaves
- Getting the solution for the particular Disease

We can make a Mobile Application and Aerial Surveillance based hardware so we can make a complete product having the above provided features so we could help the agricultural sector growth.

6 Conclusion

In this study, we have successfully accomplished in proposing a CNN for the identification of plant category and its leaf diseases by using the two online available datasets Fruit 360 [47] and Plant Village [69]. We have used various configuration for the proposed CNN including BS of 64, 128 and 256 having the DS of (70, 30), (75, 25) and (80, 20) respectively. We have achieved the best training, validating and testing accuracies of 99.95, 99.53 and 99 percent respectively. This model can be used for the identification of different Plants (Fruits/ Vegetables) categories and their leaf diseases with the good positive results apparently.

7 Appendix I

Importing and loading data from google drive

```
from google.colab import drive

drive.mount('/content/gdrive')

import os

from pydrive.auth import GoogleAuth

from pydrive.drive import GoogleDrive

from google.colab import auth

from oauth2client.client import GoogleCredentials

auth.authenticate_user()

gauth = GoogleAuth()

gauth.credentials = GoogleCredentials.get_application_default()

drive = GoogleDrive(gauth)

download = drive.CreateFile({'id': '1TmvDz74jrRFSCGdWEQo639XQnxm2AwCo'})

download.GetContentFile('Final_Dataset_fv.zip')

!unzip Final_Dataset_fv.zip

!pip install PyDrive
```

8 Appendix II

Splitting the data and getting its shape

```
datagen = ImageDataGenerator(validation_split=0.25, rescale=1./255)
train_generator = datagen.flow_from_directory(
    'Final_Dataset_fv/Training/',
    target_size=(100, 100),
    batch_size=128,
    shuffle=True,
    class_mode='categorical',
    subset='training')
validation_generator = datagen.flow_from_directory(
    'Final_Dataset_fv/Training/',
    target_size=(100, 100),
    batch_size=128,
shuffle=True,
    class_mode='categorical',
    subset='validation')
train_images,train_Labels=next(iter(train_generator))
Validation_images,Validation_Labels=next(iter(validation_generator))
#test_images,test_Labels=next(iter(test_generator))
print(train_images.shape)
print(Validation_images.shape)
#print(test_images.shape)
```

9 Appendix III

All Libraries

```
import numpy as np
from keras import layers
from keras.layers import Input, Dense, Activation, ZeroPadding2D, BatchNormalization,
Flatten, Conv2D
from keras.layers import AveragePooling2D, MaxPooling2D, Dropout, GlobalMaxPoolin
g2D, GlobalAveragePooling2D
from keras.models import Model
from keras.preprocessing import image
from keras.utils import layer_utils
from keras.utils.data_utils import get_file
from keras.applications.imagenet_utils import preprocess_input
# import cv2
#from tensorflow.python.keras.applications import ResNet50
from tensorflow.python.keras.models import Sequential
from tensorflow.python.keras.layers import Dense, Flatten, GlobalAveragePooling2D, Ba
tchNormalization
#from tensorflow.python.keras.applications.resnet50 import preprocess_input
from tensorflow.python.keras.preprocessing.image import ImageDataGenerator
from tensorflow.python.keras.preprocessing.image import load_img, img_to_array
from keras.layers import Input, Add, Dense, Activation, ZeroPadding2D, BatchNormaliz
ation, Flatten, Conv2D, AveragePooling2D, MaxPooling2D, GlobalMaxPooling2D
```

```
from keras.models import Model, load_model
from keras.preprocessing import image
from keras.utils import layer_utils
from keras.utils.data_utils import get_file
from keras.applications.imagenet_utils import preprocess_input
from IPython.display import SVG
from keras.utils.vis_utils import model_to_dot
from keras.utils import plot_model
#from resnets_utils import *
from keras.initializers import glorot_uniform
import scipy.misc
from matplotlib.pyplot import imshow
%matplotlib inline
import keras.backend as K
K.set_image_data_format('channels_last')
K.set_learning_phase(1)
from IPython.display import SVG
from keras.utils.vis_utils import model_to_dot
from keras.utils import plot_model
#from kt_utils import *
import tensorflow as tf
import matplotlib.pyplot as plt
from sklearn.metrics import precision_recall_fscore_support
```

```
from sklearn.metrics import accuracy_score
from sklearn.metrics import classification_report, confusion_matrix
from __future__ import absolute_import, division, print_function, unicode_literals
import matplotlib.pylab as plt
import tensorflow as tf
import tensorflow_hub as hub
import numpy as np
import pandas as pd
from keras import optimizers
```

10 Appendix IV

Own Multi-Layer Convolutional Neural Network building and training

```
from keras.models import Sequential
from keras.layers.normalization import BatchNormalization
from keras.layers.convolutional import Conv2D
from keras.layers.convolutional import MaxPooling2D
from keras.layers.core import Activation
from keras.layers.core import Flatten
from keras.layers.core import Dropout
from keras.layers.core import Dense
from keras import backend as K


class Own_CNN:
  @staticmethod
  def build(width, height, depth, classes):
    # initialize the model along with the input shape to be
    # "channels last" and the channels dimension itself
    model = Sequential()
    inputShape = (height, width, depth)
    chanDim = -1

    # if we are using "channels first", update the input shape
    # and channels dimension
```

```
if K.image_data_format() == "channels_first":
  inputShape = (depth, height, width)
  chanDim = 1

# CONV => RELU => POOL layer set
model.add(Conv2D(64, (3, 3), padding="valid", strides=(1,1),
  input_shape=inputShape))
model.add(Activation("relu"))
model.add(BatchNormalization(axis=chanDim))

model.add(MaxPooling2D(pool_size=(2, 2),strides=(2,2)))
#model.add(Dropout(0.2))

model.add(Conv2D(128, (3, 3), padding="valid", strides=(1,1),
  input_shape=inputShape))
model.add(Activation("relu"))
model.add(BatchNormalization(axis=chanDim))

model.add(MaxPooling2D(pool_size=(3, 3),strides=(2,2)))
#model.add(Dropout(0.2))

model.add(Conv2D(128, (3, 3), padding="valid", strides=(1,1),
  input_shape=inputShape))
```

```
model.add(Activation("relu"))
model.add(BatchNormalization(axis=chanDim))

#model.add(MaxPooling2D(pool_size=(2, 2),strides=(1,1)))
#model.add(Dropout(0.2))



model.add(Conv2D(256, (3, 3), padding="valid", strides=(1,1)))
model.add(Activation("relu"))
model.add(BatchNormalization(axis=chanDim))

#model.add(MaxPooling2D(pool_size=(2, 2),strides=(2,2)))
#model.add(Dropout(0.5))

model.add(Conv2D(256, (3, 3), padding="valid", strides=(1,1)))
model.add(Activation("relu"))
model.add(BatchNormalization(axis=chanDim))

model.add(MaxPooling2D(pool_size=(3, 3),strides=(2,2)))

# first (and only) set of FC => RELU layers
model.add(Flatten())
```

```
    model.add(Dense(2048))
    model.add(Activation("relu"))
    model.add(BatchNormalization())
    model.add(Dropout(0.5))

    model.add(Dense(2048))
    model.add(Activation("relu"))
    model.add(BatchNormalization())
    model.add(Dropout(0.5))

    model.add(Dense(512))
    model.add(Activation("relu"))
    model.add(BatchNormalization())
    model.add(Dropout(0.5))

    # softmax classifier
    model.add(Dense(classes))
    model.add(Activation("softmax"))

    # return the constructed network architecture
    return model
model = Own_CNN.build(width=100, height=100, depth=3,
  classes=len(train_generator.class_indices))
```

```
INIT_LR = 0.01

EPOCHS = 35

#BS =

print("[INFO] training network...")

opt = optimizers.SGD(lr=INIT_LR , decay=1e-4, momentum=0.9, nesterov=False)

model.compile(loss="categorical_crossentropy", optimizer=opt,

  metrics=["accuracy"])

hist=model.fit(train_generator, validation_data=validation_generator, epochs = EPOCHS

)

model.save('/content/gdrive/My Drive/Final128(75,25)_fv1_Model.h5')

shoe_model = tf.keras.models.load_model('/content/gdrive/My Drive/Final128(75,25)_fv

1_Model.h5',

custom_objects={'KerasLayer':hub.KerasLayer})
```

11 Appendix V

Getting the plots of model history, confusion matrix and classification report

```
import sys
from sklearn.metrics import confusion_matrix, classification_report
actual = []
predicted = []
np.set_printoptions(threshold=sys.maxsize)
for i in range(len(predicted_ids)):
  actual = np.append(actual, true_label_ids[i])
  predicted = np.append(predicted, predicted_ids[i])
print(confusion_matrix(actual, predicted))


print(classification_report(actual, predicted))
# summarize history for accuracy
plt.plot(hist.history['accuracy'])
plt.plot(hist.history['val_accuracy'])
plt.title('model accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()
plt.savefig('/content/gdrive/My Drive/Accuracy(70,30)_128_Plot.jpg')
# summarize history for loss
```

```
plt.plot(hist.history['loss'])

plt.plot(hist.history['val_loss'])

plt.title('model loss')

plt.ylabel('loss')

plt.xlabel('epoch')

plt.legend(['train', 'test'], loc='upper left')

plt.show()

plt.savefig('/content/gdrive/My Drive/Loss(70,30)_128_Plot.jpg')
```

12 Appendix VI

Testing the images

```
from PIL import Image
import numpy as np
from skimage import transform
def load(filename):
   np_image = Image.open(filename)
   np_image = np.array(np_image).astype('float32')/255
   np_image = transform.resize(np_image, (100, 100, 3))
   np_image = np.expand_dims(np_image, axis=0)
   return np_image

image = load('Final_Dataset_fv/Test Images/Apple___Cedar_apple_rust_Leaf.jpg')
tf_model_predictions = shoe_model.predict(image)
predicted_ids = np.argmax(tf_model_predictions, axis=-1)
print(predicted_ids)
predicted_labels = dataset_labels[predicted_ids]
print(predicted_labels)
```

13 Appendix VII

Data Augmentation and resizing

```
import matplotlib.pyplot as plt

import os

import PIL

from PIL import Image

import cv2

from PIL import ImageEnhance

input_dir = '/content/Fruits Dataset/Training/'

output_dir='/Users/zaid/Desktop/Disease_Dataset/Testing/'

x=os.listdir(input_dir)

print(x)

i=1

y=os.listdir(input_dir+x[0])

#for image in os.listdir(input_dir+x[0]):

for x in os.listdir(input_dir):

  i=1

  for image in os.listdir(input_dir+x):

    img=Image.open(input_dir+x+"/"+image)

    enhancer = ImageEnhance.Sharpness(img)

    brightner=ImageEnhance.Brightness(img)

    brght_img=brightner.enhance(1.5)

    enhnc_img=enhancer.enhance(3)
```

```
    hflip_img = img.transpose(PIL.Image.FLIP_TOP_BOTTOM)
    '''if not os.path.exists(input_dir+x+"2"):
      os.mkdir(input_dir+x+"2")'''
    #rot_img_90.save(input_dir+x+"2"+"/"+"rot_img_90_"+str(i)+".jpg")
    enhnc_img.save(input_dir+x+"/"+"enhnc_img_"+image)
    brght_img.save(input_dir+x+"/"+"brght_img_"+image)
    hflip_img.save(input_dir+x+"/"+"hflip_img_"+image)
    i+=1
    #plt.imshow(img)
# Resizing
import os
import cv2
input_dir = '/Users/zaid/Desktop/DiseaseDataset/Testing/'
output_dir='/Users/zaid/Desktop/Disease_Dataset/Testing/'
i=0
for x in os.walk(input_dir):
    for a in x[1]:
        print(a)
        if not os.path.exists(output_dir+a+"_Leaf"):
            os.mkdir(output_dir+a+"_Leaf")
        try:
            i=0
            print(a+"_Leaf")
```

```
    for file in os.listdir(input_dir+a):
        path=input_dir+a+"/"+file
        img=cv2.imread(path,cv2.IMREAD_UNCHANGED)
        image = cv2.resize(img, (100,100), interpolation = cv2.INTER_AREA)
        cv2.imwrite(output_dir+a+"_Leaf/"+str(i)+'_200.jpg',image)
        i+=1
    print(i)
except OSError:
    print('file not found')
```

14 References

[1] R. A. Castaneda Aguilar, D. T. T. Doan, D. L. Newhouse, M. C. Nguyen, H. Uematsu and J. P. W. D. Azevedo, "Who are the poor in the developing world ?," *Policy Research working paper,* no. WPS 7844, 2016.

[2] J. Alston and P. Pardey, "Agriculture in the Global Economy," *The Journal of Economic Perspectives,* vol. 28, 2014.

[3] "Agriculture Overview," The World Bank, 23 September 2019. [Online]. Available: https://www.worldbank.org/en/topic/agriculture/overview. [Accessed 22 February 2020].

[4] Alston, J. M., M. A. Andersen, J. S. James and a. P. G. Pardey, Persistence Pays: U.S. Agricultural Productivity Growth and the Benefits from Public R&D Spending, Springer-Verlag New York, 2010.

[5] "Importance of Agriculture Sector in a Country's Economic Development," IPPMedia, 02 December 2016. [Online]. Available: https://www.ippmedia.com/en/features/importance-agricultural-sector-country%E2%80%99s-economic-development.

[6] FEOSTAT, "Food and Agriculture Organisation," World Bank, February 2020. [Online]. Available: http://www.fao.org/faostat/en/#data/QC.

[7] K. Golhani, S. K. Balasundram, G. Vadamalai and B. Pradhan, "A review of neural networks in plant disease detection using hyperspectral data," *Information Processing in Agriculture,* vol. 5, no. 3, pp. 354-371, 2018.

[8] Y. Toda and F. Okura, "How Convolutional Neural Networks Diagnose Plant Disease," *Plant Phenomics,* vol. 2019, p. 14, 2019.

[9] M. Bag, M. Yadav and A. Mukherjee, "Changing Disease Scenario with Special Emphasis on False Smut of Rice," *SATSA Mukhapatra,* no. 21, pp. 219-224, 2012.

[10] J. Zadoks, "On the political economy of plant disease epidemics: Capita selecta in historical epidemiology," *Wageningen Academic Pub,* pp. 1-249, 2008.

[11] C. P. Wild and Y. Y. Gong, "Mycotoxins and human disease: a largely ignored global health issue," *Carcinogenesis,* vol. 31, no. 1, pp. 71-82, 2010.

[12] S. Savary, A. Ficke, J.-N. Aubertot and C. Hollier, "Crop losses due to diseases and their implications for global food production losses and food security," *Food Security. 4,* pp. 519-537, 2012.

[13] R. Balodi, S. Bisht, A. Ghatak and K. Rao, "Plant Disease Diagnosis: Technological Advancements and Challenges," *Indian Phytopathology,* vol. 70, pp. 275-281, 2017.

[14] J. Ma, K. Du, F. Zheng, L. Zhang, Z. Gong and Z. Sun, "A recognition method for cucumber diseases using leaf symptom images based on deep convolutional neural network," *Computers and Electronics in Agriculture,* vol. 154, pp. 18-24, 2018.

[15] S. Chouhan, A. Koul, D. U. Singh and S. Jain, "Bacterial foraging optimization based Radial Basis Function Neural Network (BRBFNN) for

identification and classification of plant leaf diseases: An automatic approach towards Plant Pathology," *IEEE Access,* vol. 6, pp. 8852-8863, 2018.

[16] Z. Iqbal, M. Khan, M. Sharif and J. Shah, "An automated detection and classification of citrus plant diseases using image processing techniques: A review," *Computers and Electronics in Agriculture,* vol. 153, pp. 12-32, 2018.

[17] "Plant pests and diseases," Food And Agriculture Organisation of the United Nations, 2012. [Online]. Available: http://www.fao.org/emergencies/emergency-types/plant-pests-and-diseases/en/.

[18] A. van den Oord, S. Dieleman and B. Schrauwen, "Deep content-based music recommendation," *Advances in Neural Information Processing Systems,* pp. 2643-2651, 2013.

[19] Q. Hardy, "Reasons to Believe the A.I. Boom is Real," The New York Times, 18 07 2016. [Online]. Available: https://www.nytimes.com/2016/07/19/technology/reasons-to-believe-the-ai-boom-is-real.html.

[20] U. Mokhtar, N. E. Bendary, A. E. Hassenian, E. Emary, M. A. Mahmoud, H. Hefny and M. F. Tolba, "SVM-Based Detection of Tomato Leaves Diseases," *Advances in Intelligent Systems and Computing,* vol. 323, 2015.

[21] S. Mohanty, D. Hughes and M. Salathe, "Using Deep Learning for Image-Based Plant Disease Detection," *Frontiers in Plant Science,* vol. 7, 2016.

[22] J. Amara, B. Bouaziz and A. Algergawy, "A Deep Learning-based Approach for Banana Leaf Diseases Classification," 2017.

[23] U. P. Singh, S. S. Chouhan, S. Jain and S. Jain, "Multilayer Convolution Neural Network for the Classification of Mango Leaves Infected by Anthracnose Disease," *IEEE Access,* vol. 7, pp. 43721-43729, 2019.

[24] B. Liu, Y. Zhang, D. He and Y. Li, "Identification of Apple Leaf Diseases Based on Deep Convolutional Neural Networks," *Symmetry,* vol. 10, p. 11, 2017.

[25] "AgriculturalPost.com," [Online]. Available: https://agriculturepost.com/artificial-intelligence-in-agriculture-eases-risk-of-changing-weather-patterns/. [Accessed Monday January 2021].

[26] M. H. P. Kanjalkar and P. S.S.Lokhande, "Detection and Classification of Plant Leaf Diseases using ANN," *International Journal of Scientific & Engineering Research,* vol. 4, pp. 1777-1780, 2013.

[27] S. Sladojevic, M. Arsenovic, A. Anderla, D. Culibrk and D. Stefanovic, "Deep neural networks based recognition of plant diseases by leaf image classification," *Computational intelligence and neuroscience 2016,* vol. 2016, p. 11, 2016.

[28] M. Brahimi, K. Boukhalfa and A. Moussaoui, "Deep learning for tomato diseases: classification and symptoms visualization," *Applied Artificial Intelligence,* vol. 31(4), pp. 299-315, 2017.

[29] C. DeChant, T. Wiesner-Hanks, S. Chen, E. L. Stewart, J. Yosinski, M. A. Gore, R. J. Nelson and H. Lipson, "Automated identification of northern leaf blight-infected maize plants from field imagery using deep learning," *Phytopathology,* vol. 107(11), pp. 1426-1432, 2017.

[30] Y. Lu, S. Yi, N. Zeng, Y. Liu and Y. Zhang, "Identification of rice diseases using deep convolutional neural networks," *Neurocomputing,* vol. 267, pp. 378-384, 2017.

[31] G. E. Hinton, "A practical guide to training restricted Boltzmann machines," *Neural networks: Tricks of the trade,* pp. 599-619, 2012.

[32] S. H. Lee, C. S. Chan, S. J. Mayo and P. Remagnino, " How deep learning extracts and learns leaf features for plant classification," *Pattern Recognition,* vol. 71, pp. 1-13, 2017.

[33] A. Ramcharan, K. Baranowsk, P. McCloskey, B. Ahmed, J. Legg and D. P. Hughes, "Deep learning for image-based cassava disease detection," *Frontiers in plant science,* vol. 8, p. 1852, 2017.

[34] A. Fuentes, S. Yoon, S. C. Kim and D. S. Park, "A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition," *Sensors,* vol. 17(9), p. 2022, 2017.

[35] G. Wang, Y. Sun and J. Wang, "Automatic image-based plant disease severity estimation using deep learning," *Computational intelligence and neuroscience,* vol. 2017, 2017.

[36] Y. D. Zhang, Z. Dong, X. Chen, W. Jia, S. Du, K. Muhammad and S. H. Wang, "Image based fruit category classification by 13-layer deep convolutional neural network and data augmentation," *Multimedia Tools and Applications,* vol. 78(3), pp. 3613-3632, 2019.

[37] A. K. Rangarajan, R. Purushothaman and A. Ramesh, "Tomato crop disease classification using pre-trained deep learning algorithm," *Procedia computer science* , vol. 133, pp. 1040-1047, 2018.

[38] I. Hussain, Q. He and Z. Chen, "Automatic fruit recognition based on DCNN for commercial source trace system," *International Journal on Computational Science and Application (IJCSA),* vol. 8, 2018.

[39] X. Zhang, Y. Qiao, F. Meng, C. Fan and M. Zhang, "Identification of maize leaf diseases using improved deep convolutional neural networks," *IEEE Access,* vol. 6, pp. 30370-30377, 2018.

[40] B. A. M. Ashqar and S. S. Abu-Naser, "Image-based tomato leaves diseases detection using deep learning," *International Journal of Engineering Research,* vol. 2, pp. 10-16, 2018.

[41] J. G. Barbedo, "Factors influencing the use of deep learning for plant disease recognition," *Biosystems engineering,* vol. 172, pp. 84-91, 2018.

[42] K. P. Ferentinos, "Deep learning models for plant disease detection and diagnosis," *Computers and Electronics in Agriculture* , vol. 145, pp. 311-318, 2018.

[43] D. Oppenheim, G. Shani, O. Erlich and L. Tsror, "Using deep learning for image-based potato tuber disease detection," *Phytopathology,* vol. 109(6), p. 10831087, 2019.

[44] M. Pineda, M. L. Pérez-Bueno and M. Barón, "Detection of bacterial infection in melon plants by classification methods based on imaging data," *Frontiers in plant science,* vol. 9, p. 164, 2018.

[45] E. C. Too, L. Yujian, S. Njuki and L. Yingchun, " A comparative study of fine-tuning deep learning models for plant disease identification.," *Computers and Electronics in Agriculture,* vol. 161, pp. 272-279, 2018.

[46] S. W. H. a. C. Z. Zhang, "Three-channel convolutional neural networks for vegetable leaf disease recognition," *Cognitive Systems Research,* vol. 53, pp. 31-41, 2019.

[47] H. Muresan and M. Oltean, "Fruit recognition from images using deep learning," *Acta Universitatis Sapientiae, Informatica,* vol. 10(1), pp. 26-42, 2018.

[48] J. G. A. Barbedo, "Plant disease identification from individual lesions and spots using deep learning," *Biosystems Engineering,* vol. 180, pp. 96-107, 2019.

[49] A. Kaya, A. S. Keceli, C. Catal, H. Y. Yalic, H. Temucin and B. Tekinerdogan, "Analysis of transfer learning for deep neural network based plant," *Computers and electronics in agriculture,* vol. 158, pp. 20-29, 2019.

[50] S. Zhang, S. Zhang, C. Zhang, X. Wang and Y. Shi, "Cucumber leaf disease identification with global pooling dilated convolutional neural network," *Computers and Electronics in Agriculture,* vol. 162, pp. 422-430, 2019.

[51] G. Geetharamani and A. Pandian, "Identification of plant leaf diseases using a nine-layer deep convolutional neural network," *Computers & Electrical Engineering,* vol. 76, pp. 323-338, 2019.

[52] P. Jiang, Y. Chen, B. Liu, D. He and C. Liang, "Real-Time Detection of Apple Leaf Diseases Using Deep Learning Approach Based on Improved Convolutional Neural Networks," *IEEE Access,* vol. 7, pp. 59069-59080, 2019.

[53] Y. Toda and F. Okura, "How Convolutional Neural Networks Diagnose Plant Disease," *Plant Phenomics,* vol. 2019, 2019.

[54] M. R. Mia, S. Roy, S. K. Das and M. A. Rahman, "Mango Leaf Diseases Recognition Using Neural Network and Support Vector Machine," *International Journal of Computer Science Engineering,* vol. 8, 2019.

[55] A. Blessy and D. J. W. Wise, "Detection of Affected Part of Plant Leaves and Classification of Diseases Using CNN Technique," *International Journal of Engineering and Techniques,* vol. 4, no. 2, 2018.

[56] S. Sunkad, A. Bagale, S. Rumale and Z. Shariff, "Multiclass Classification Method Based On Deep Learning For Leaf Identification For Farmers," *IRJET,* vol. 3(5), 2019.

[57] M. Arsenovic, M. Karanovic, S. Sladojevic, A. Anderla and D. Stefanovic, "Solving Current Limitations of Deep Learning Based approaches for Plant Disease Detection," *Symmetry,* vol. 11(7), p. 939, 2019.

[58] L. Deng and D. Yu, "Deep Learning: Methods and Applications," *Foundations and Trends® in Signal Processing,* vol. 7, pp. 197-387, 2014.

[59] R. Dechter, "Learning While Searching in Constraint-Satisfaction-Problems," *AAAI,* pp. 178-185, 1986.

[60] Y. LeCun, L. Bottou, Y. Bengio and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE,* vol. 86(11), pp. 2278-2324, 1998.

[61] A. Krizhevsky, I. Sutskever and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems,* pp. 1097-1105, 2012.

[62] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," *European conference on computer vision,* pp. 818-833, 2014.

[63] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke and A. Rabinovich, "Going Deeper with Convolutions," in *In Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015.

[64] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *arXiv 1409.1556,* 2014.

[65] K. He, X. Zhang, S. Ren and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.

[66] S. Saha, "Towards Data Science," [Online]. Available: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53. [Accessed Monday January 2021].

[67] J. Murphy, "An Overview of Convolutional Neural Network Architectures for Deep Learning," *Microway, Inc*, p. 22, 2016.

[68] "Wikipedia," [Online]. Available: https://en.wikipedia.org/wiki/F-score. [Accessed 25 November 2020].

[69] M. S. David P. Hughes, "An open access repository of images on plant health to enable the development of mobile disease diagnostics," *arvix.org*, 2016.

[70] S. Saha, "Towards Data Science," [Online]. Available: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53. [Accessed Monday January 2021].

Proposed Certificate for Plagiarism

It is certified that MS Thesis Titled **Multi-Layer Convolutional Neural Network Based Identification of Plant Category and Leaf Diseases** by **Hammas Bin N** has been examined by us. We undertake the follows:

a. Thesis has significant new work/knowledge as compared already published or are under consideration to be published elsewhere. No sentence, equation, diagram, table, paragraph or section has been copied verbatim from previous work unless it is placed under quotation marks and duly referenced.
b. The work presented is original and own work of the author (i.e. there is no plagiarism). No ideas, processes, results or words of others have been presented as Author own work.
c. There is no fabrication of data or results which have been compiled/analyzed.
d. There is no falsification by manipulating research materials, equipment or processes, or changing or omitting data or results such that the research is not accurately represented in the research record.
e. The thesis has been checked using TURNITIN (copy of originality report attached) and found within limits as per HEC plagiarism Policy and instructions issued from time to time.

Name & Signature of Supervisor

Dr. Muhammad Jawad Khan

Signature : ________________

Multi-Layer Convolutional Neural Network Based Identification of Plant Category and Leaf Diseases

ORIGINALITY REPORT

13%	7%	9%	4%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Higher Education Commission Pakistan Student Paper	1%
2	Julian M. Alston, Philip G. Pardey. "Agriculture in the Global Economy", Journal of Economic Perspectives, 2014 Publication	1%
3	www.ijert.org Internet Source	<1%
4	medium.com Internet Source	<1%
5	Uday Pratap Singh, Siddharth Singh Chouhan, Sukirty Jain, Sanjeev Jain. "Multilayer Convolution Neural Network for the Classification of Mango Leaves Infected by Anthracnose Disease", IEEE Access, 2019 Publication	<1%
6	onlinelibrary.wiley.com Internet Source	<1%
7	www.coursehero.com Internet Source	<1%