Fruits Category Identification and Infected Part Detection using Deep Learning

Author

ZAID ISMAIL

Regn Number

00000206777

Supervisor

Dr. M. Jawad Khan

ROBOTICS AND INTELLIGENT MACHINE ENGINEERING

SCHOOL OF MECHANICAL & MANUFACTURING ENGINEERING

NATIONAL UNIVERSITY OF SCIENCES AND TECHNOLOGY

ISLAMABAD

JUNE 2021

Fruits Category Identification and Infected Part Detection using Deep Learning

Author

ZAID ISMAIL

Regn Number

00000206777

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

JUNE, 2021

Declaration

I certify that this research work titled "*Fruits Category Identification and Infected Part Detection using Deep Learning"* 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

ZAID ISMAIL

00000206777

Plagiarism Certificate (Turnitin Report)

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

Signature of Student

ZAID ISMAIL

00000206777

Signature of Supervisor

Copyright Statement

Acknowledgements

To begin, I want to express my heartfelt gratitude to Allah the Almighty for all the blessings that have come my way throughout my life, particularly during my educational career. Allah has always guided me in the direction that was best for me in every way. Pursuing my master's degree and completing the required course work and research would not be possible without Almighty Allah's guidance.

I'd like to express my heartfelt gratitude to my Thesis Supervisor, "Dr. M. Jawad Khan," for believing in me and allowing me to work with him. His guidance, assistance, and availability throughout my research phase enabled me to complete this work. I'd like to express my gratitude to him for guiding me through all of the events other than my research, which have aided me in honing my technical skills and remaining motivated. His efforts and dedication throughout my research work were instrumental in achieving this goal.

I would like to express my gratitude to Dr. Hasan Sajid, Dr. Omer Ansari, and Dr. Yasar Ayaz of my committee foar their encouragement and support throughout my course work and research.

I am extremely appreciative of my parents for supporting and loving me throughout my life to the best of their abilities. Especially my father, Muhammad Ismail, who has always served as a role model for me by demonstrating the value of hard work, honesty, dedication, and commitment in all aspects of life.

Additionally, I'd like to express my gratitude to my friends for their encouragement and motivation, specially wants to thank Miss. Saima Jamil, Mr. Hammas Bin Naeem and Mr. Farhan Hassan for his availability and help throughout my Master’s degree.

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

Abstract

Agriculture Sector is responsible for providing a living or sustainable income to the majority of the population in the world. In Pakistan, agricultural plays a very vital role for the development and stability of the economy. Agriculture is the major source of providing food to every house as well as raw material for the development of industrial sector. Approximately 48 percent of the labor force is directly engaged with this sector. Agriculture provides numerous advantages to the population as well as it is affected by various aspects which eventually leads in effecting the overall sectors yield as well as the economy of the country on bigger scale. Various diseases in the crop is one of the leading factors which is responsible for the destruction of overall yield and the economy and it is very difficult for a naked eye to keep information of every type of crop disease and identifying the infected part of the crop. It is so important to take necessary measures for the automated solution to this problem so we could produce good quality yield and benefit the overall economy. Computer Vision and Deep Learning is one of the most utilized recent technological fields in every domain of daily life. With the advancement in computer with high quality processing these fields have made a big name in wide area of applications. We have seen so much applications of these fields in agriculture sector also for example, Crop Disease Identification, Disease Detection, Disease Area Detection etc. Many algorithms including YOLO had reviewed for this study. This thesis proposes a DL method for the classification of the fresh and rotten fruits and the category of the fruit using the CNN. In this study we have also detected the rotten part of the fruit using the YOLOV5. Dataset was calculated by two online available datasets “Fruits 360” and “Fruits fresh and rotten for classification”. Two classes of fruits from “Fruits 360” dataset were added in the “Fruits fresh and rotten for classification” dataset. Similarly, the images of two classes of fruits were captured

through mobile phone and added in the “Fruits fresh and rotten for classification” dataset. The complete dataset contains 18k images approximately. For, detecting the rotten part of the fruit, some of the images from all the rotten fruits classes were taken and annotated to apply YOLOV5. The dataset for YOLOV5 contains 1000 annotated images approximately. Our model has achieved best possible results for classification of various categories and detection of rotten part of the fruits.

Keywords: *Gross Domestic Product, Artificial Intelligence, Computer Vision, Machine Learning, Deep Learning, Convolutional Neural Network, YOLOV5, MobileNetV2, InceptionV3, VGG16*

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
You Only Look Once	YOLO

1 Introduction

Agriculture plays a critical role in the growth and development of a Country's Economy. It is the backbone of the global economy, the largest provider of food, and is responsible for providing a sustainable income to millions of households worldwide. According to 2016 analysis, 65% of poor working adults make their living from agriculture [1]. The region accounts for a comparatively minor portion of the global economy. Agriculture accounted for just 2.8 percent of the global economy, according to a 2012 World Bank report. To put it bluntly, agriculture accounts for 19 percent (1.3 billion) of human capital [2]. Interestingly, agriculture accounted for a whopping one-third of global gross domestic product (GDP) in 2014 [3]. Agriculture, when compared to other sectors, has the potential to increase the incomes of the poorest by up to fourfold. This is because agriculture plays a larger role in low- and middle-income countries than it does in high-income countries. For example, agriculture accounts for 18% of total national income and 54% of employment in India [4]. Agriculture has shifted over the last half-century from being a major source of income for high-income countries through the use of superior technologies to being a major source of income for low- and middle-income countries in Asia and the Pacific. For instance, in 1961, countries such as the United States produced approximately 43% of total agricultural sector output, compared to 24.6 percent in 2011. Asia and the Pacific increased its share of the global economy from 23.9 percent in 1961 to 44.7 percent in 2011 [2]. Several possible explanations include the rapid urbanisation and industrialisation of developed countries, as well as the increasing pressure on developing countries to accommodate their rapidly growing populations [5].

40% of the world's land area is used for agricultural purposes, and only 0.78 percent of the world's cropland comes from the 100 least-contributing countries between 2009 and 2011. India, the United States, the Russian Federation, China, and Brazil account for 42.1 percent of total cropland; nearly three-quarters of total world output comes from the top 20 countries, with China dominating almost every major crop at 23.0 percent of total agricultural output, followed by the United States at 10.1 percent.

The global population is increasing at a breakneck pace, estimated to reach 9.1 billion by 2050 (FAO), necessitating an additional 70% increase in food production [6]. Thus, sustainable agricultural growth and protection are critical for meeting the world's food requirements in the coming decades and will play a critical role in the global economy's poverty alleviation.

1.1 Background

Fruit rotting has substantial economic consequences; it is estimated that roughly a third of the cost of fruit is spent on rotting materials. Furthermore, fruit sales will be impacted since customers believe that rotten fruits are harmful to their health [7] as reduced concentrations of amino acids, vitamins, sugar/glucose, and other nutrients unavoidably create public worries about edibility difficulties, prompting discussions about how to halt or reduce the decaying process.

Fruit quality categorization is significant because of the importance of food status in people's lives and its contribution to the economy, yet it is a time-consuming manual activity. Automation of categorization through the use of digital methods is thought to be the answer to this problem.

Furthermore, a shortage of nutrients can produce difficulties that lead to the development of black spots, such as insufficient calcium causing apples to develop cork spots [8]. Another component is oxygen exposure, as an enzyme called polyphenol oxidase (PPO) initiates a cascade of biochemical events involving proteins, pigments, fatty acids, and lipids that cause the fruit to fade in colour and degrade in taste and fragrance [9].

When fruit deteriorates, established research evidence demonstrates that it goes through a sequence of metabolic transformations that result in changes in its physical conditions, such as colour and shape. The majority of these characteristics can be recorded. The most cost-effective solution is projected to be a computer vision-based technique. Computer vision technology is used in automatic categorizing of fruits and vegetables [10].

Detection of illnesses and deformities in fruits and vegetables induced by alien biological invasion [11], fruit/vegetable classification for a variety of horticulture goods [12]. Fruit/vegetable object real-time tracking [13], and so on.

The visual aspects of fruit rot have been observed by academics in this field [14]. Polysaccharides cellulose, hemicellulose, and pectin are the most commonly encountered chemical components for cell construction, and starch is the principal storage polymer. Microbe invasion occurs when these polymers are corrupted by cellular lytic enzymes, which take nutrients (water and other intracellular elements) for growth. Fruits have protective epidermal barriers to ward off invaders, which are usually covered by a waxy cuticle layer that gives them its natural glisters [15]. Spoilage is a significant degradation sign that causes fruits to shrivel due to a lack of cell fluid.

Another essential metric for determining how much a fruit has deteriorated is texture, which depicts the typical properties of fruit surfaces. Turgor pressure and plant cell lamella, which binds individual cells together, determine the texture of a healthy fruit [16]. Spoilage can cause cell deformation and disintegration, resulting in a morphological change from overall roughness to wizened surfaces.

Visual object identification makes considerable use of deep learning. For fruit and vegetable recognition, the study [17]. In [18] YOLO is quick compared to other techniques, achieving a real-time image processing speed of 20 frames per second. Another study [19] uses a deep neural network VGG for fruit recognition, demonstrating that convolutional neural networks can reach high accuracy as they go deep.

The disease penicillium digitatum affects the decay process in mandarins, and there is research [20] dedicated to early diagnosis of this disease by evaluating decay visual aspects. A collection of decision trees captures and processes the visual features. These trials, however, are limited to one type of fruit and assume no background noise. The grading mechanism is a classification model that considers fruit as either healthy or rotten/defective.

The efficiency of picture classification tasks has improved because to recent advances in computer vision, particularly in the disciplines of machine learning and deep learning [20-25]. One of the major challenges in the agricultural fields is the detection of defective fruits and the classification of fresh and rotten fruits. If not properly identified, rotten fruits can cause damage to other fresh fruits and reduce productivity. Traditionally, men performed this classification, which was a labor-intensive, time-consuming, and inefficient practice. Furthermore, it increases the cost of production. As a result, we require an automated system that can reduce human effort, increase productivity, and minimize production cost and time.

As a result, not only is the detection, identification, and diagnosis system complex, but it is also costly and time intensive [26]. Farmers are frequently unable to notice or identify a problem in their field in a timely manner, rendering enormous swaths of crops unusable. Diseases and illnesses are the most common causes of plant death. Diseases are caused by external biotic assault by bacteria or fungi, whereas disorders are caused by abiotic conditions such as temperature, rainfall, and soil health [27]. This drop in yield has far-reaching economic consequences, both in terms of money and food and nutrition security. Early detection, diagnosis, and eradication of disease infestations are critical for avoiding yield loss. This will prevent the illness from spreading to other parts of the field, resulting in considerable quantitative and qualitative yield loss.

The SNARC, the first neural network system capable of learning, was invented by Marvin Minsky and Dean Edmonds in 1951, effectively giving birth to neural networks. Frank Rosenblatt produced The Perceptron, the first artificial neural network (ANN), in 1958, which inspired the convolutional neural network. Due to a number of intrinsic design problems, the technology initially failed to gain traction. The main problems were that it lacked sound theory, that neural networks required a lot of data, that it required a lot of computing power, and that trained models tended to overfit. As a result, after the publication of work on SVMs by Corinna Cortes and Vladmir Vapnik in the late 1990s, Support Vector Machines (SVMs) took over the scene.

They have become the de facto approach for visual imagery analysis due to the advent of personal computers, their ability to execute enormous computations, and a better knowledge of CNNs. CNNs have increased in popularity over the previous decade, and they continue to do so as a means of detecting and classifying images and videos, as well as recommendation systems [28] and image analysis. CNNs are a deep learning subset made up completely of fully connected multilayer perceptrons (A class of ANNs). Popularity can also be attributed to visual databases like ImageNet and Kaggle, which supply the required data for CNN models to be trained, hence acting as a catalyst for AI's continuing growth [29] .

Significant progress has been made in the fields of Computer Vision (CV), Machine Learning (ML), and Artificial Intelligence (AI) since their beginning. These three fields are constantly producing new work and are rapidly expanding. In all sectors, the development of automated models is accelerating, integrating the strengths of these three fields of research. In agriculture, Deep Learning (DL) techniques (a subset of machine learning algorithms with numerous layers) are used to identify and recognise fruits and agricultural diseases. The technology available at the time hampered the development of DL, much as it did with ANNs (late 1990s). Its debut into the technological area was hampered by a lack of processing capacity and a scarcity of huge data sets. In recent years, DL's popularity has risen because to significant cost reductions and increasing availability of solid data. Deep learning has progressed faster because to specialised data sets like Kaggle and ImageNet. These databases not only store vast amounts of information, but they also conduct annual competitions to encourage academics and scientists. These challenges led to the development of more accurate neural networks, such as AlexNet, created by Alex Krizhevsky in 2012. The mistake rate of AlexNet was cut in half compared to the previous year's winner.

Numerous difficult and time-consuming problems have been solved using ANNs, CNNs, and deep learning. For example, identifying and classifying leaves and leaf diseases fresh and rotten fruits [30].

This research [31] provided a new method for detecting fruit skin problems utilising a machine vision system that has been shown to be more accurate, more resistant to colour noise, and with lower calculation costs. The colour histogram is retrieved as an image feature in the local picture patch, and the Linear SVM (Support vector machine) is utilised to learn the model. In the case of orange inspection, this method achieves a 96.7 percent recall rate and a 1.7 percent false detection rate.

In this study [39], a Convolutional Neural Network has been used, which is a type of deep learning architecture (CNN). As a result, we employ CCN to detect mangosteen. CNN has proven to be extremely effective at classifying photos. To ensure data accuracy, this CNN approach employs a 4-fold Validation Cross. Initializing the parameter settings during the creation of the CNN architecture model speeds up the network training procedure. The results of the studies employing the CNN algorithm revealed that fault identification on the mangosteen fruit performed at a rate of 97 percent.

This research aimed [32] to develop an automated and efficient method for detecting apple flaws. Based on laser-induced light backscattering imaging and the convolutional neural network (CNN) technique, an automatic detection approach for apple flaws is provided. A semiconductor laser is used to create laser backscattering spectroscopic images of apples. To obtain the best image dataset for CNN, we do preprocessing steps. To detect apple faults, an AlexNet model with an 11-layer structure was created and trained. They have examined how effectively the model recognises the consequences of apple flaws. The suggested CNN model for detecting apple faults has a 92.5 percent identification rate and is more accurate than traditional machine learning techniques.

Figure 1 AI in Agriculture

1.2 Problem Statement

Agriculture is critical to the economy of the country, as the globe advances with significant technical improvements in all fields, including agriculture. Various diseases in crops are difficult to detect with the naked eye, and it is extremely difficult for a human to maintain track of a large number of minor diseases that have an impact on crop productivity. The goal of this project is to create an effective Multi-Layer Convolutional Neural Network for recognizing diverse fruits. In the agricultural industry, detecting rotting fruits has become crucial. The classification of fresh and rotten fruits is usually carried out by people, which is ineffective for fruit farmers. Humans feel fatigued after performing the same work repeatedly, whereas machines do not. As a result, the study presents a method for reducing human effort, cost, and production time in the agricultural industry by recognizing faults in the fruits. If the faults are not detected, the contaminated fruits may contaminate the good fruits. As a result, we have adopted two methods:

A. Classification Method:

In our work, we have created dataset of 5 Fresh and Rotten Fruits (10 classes) using publicly available datasets "Fruits Fresh and Rotten for Classification" and "Fruit 360" from KAGGLE. We have obtained two classes from Fruit 360 and captured two classes using mobile and merged with "Fresh and Rotten for Classification dataset" and then applied Transfer learning using three pre-trained models to get results on our dataset.

- MobileNetV2
- VGG16

- InceptionV3

B. Detection Method:

We have used the pre-trained weights of the YOLO V5 model to detect the rotten part of the fruits.

We have annotated the images of the rotten fruits using sense.ai and then apply YOLOV5 for the detection of the infected part of the fruit.

1.3 Objectives

- To collect data from various sources.
- To augment and annotate data.
- To Identify the fruit category using Convolutional Neural Networks (CNNs).
- To detect the infected part of the rotten fruits.
- To assist the agricultural sector by providing an Intelligent solution for fresh and rotten fruit classification.

1.4 Areas of Application

Following are the major areas of application of this work

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

1.5 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

Visual object identification makes considerable use of deep learning. For fruit and vegetable recognition, the study [33]. In [34] YOLO is quick compared to other techniques, achieving a real-time image processing speed of 20 frames per second. The fruits in the project, on the other hand, are constrained by the settings in which they remain connected to their biological hosts.

2.1.1 YOLO:

You Only Look Once (YOLO) is a phrase that means "You Only Look Once" [35]. The object anchoring procedure is treated as a regression problem by YOLO, which requires the anchor coordinate, width ww, and height h to be defined for object localization. One of the advantages of YOLO over other CNN techniques is that it considers global input rather than local information. The input photos are divided into a grid consisting of an SS × SS grid of cells by YOLO. If a cell has a piece of an object, the cell is in charge of detecting it. Each cell generates BB bounding boxes and confidence scores in the same way. The model's confidence level in the bounding box enclosing the target item is described by confidence scores. The confidence can be defined as in the eq. (1)

$$Prob(Object) \times IOU_{predict}^{truth} \quad \text{……………….} (1)$$

where IOU (Intersection Over Union) is a process for calculating the overlapping area of two unions. In this case, the IOU should be the intersection between the ground truth and the predict. The resultant bounding box should be the shared area of the two unions. $Prob(Object)$ refers to whether the grid cell contains an object or not.

Consider that each object should have a label, the confidence can be expressed as eq. (2) for the prediction of a bounding box encapsulating an object of a class

$$Prob(class_i) \times IOU_{predict}^{truth} = Prob(class_i \mid Object) \times Prob(Object) \times IOU_{predict}^{truth}.$$
……(2)

As a result, each cell should predict a total of five parameters. The four parameters that define a bounding box are location and size $(xx,,,h)$. The probability of each class is associated with the detected object.

In real implementation, a particular YOLO [36] proposed by the author has 24 convolutional layers followed by two dense layers. YOLO employs 1×1 reduction layers with 3×3 convolutional layers following behind. This structure is shown in Figure 2.

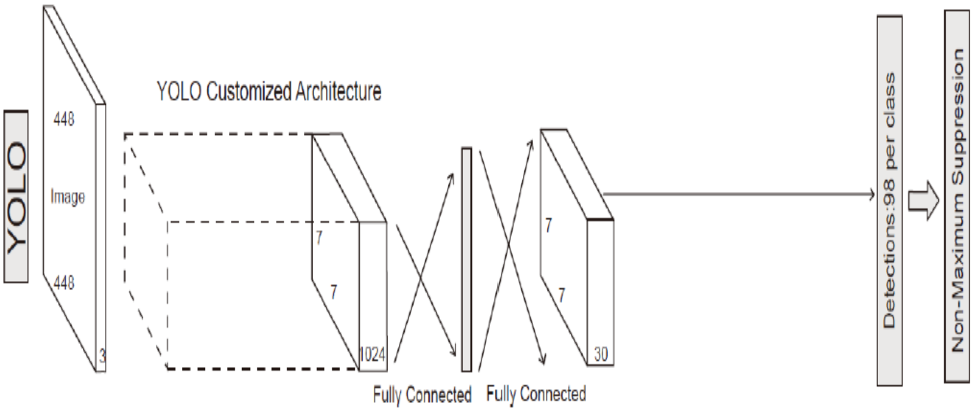

Figure 2: YOLO architecture

2.2 Transfer Learning

In this study we have applied Transfer Learning using the following three pre-trained models.

- VGG16
- InceptionV3
- MobileNetV2

2.2.1 VGG16 Architecture

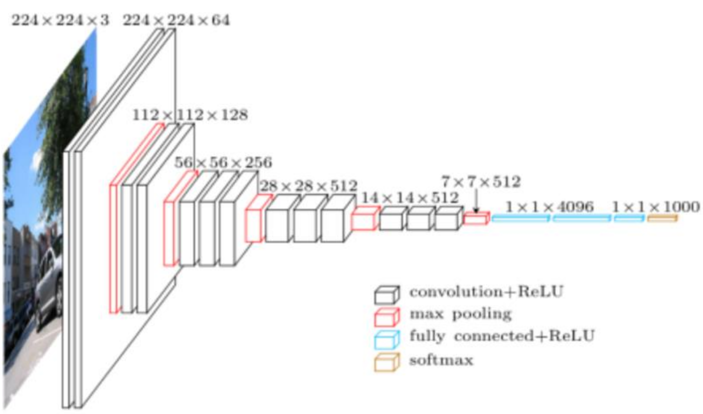

Figure 3: VGG16 Architecture

2.2.1.1 Configuration details of VGG16

Convolutional Layers	13 (each with 3x3 Filters and stride 1)
Max Pooling	5 (with 2x2 and stride 2)
Fully Connected Layers	2 (Units = 4096)
Output Layer	1(no.of classes, activation = “Softmax”)
Learning Rate	0.0001
Weight Decay	1e-4
Activation Function	ReLU
Batch Size	32
Epochs	50
Training Algorithm	ADAM

Table 1: VGG16 Configuration

2.2.2 InceptionV3 Architecture

Ramcharan et al. [38] used the Inception v3 network in 2017 to discover three cassava diseases and two insect pests. Brown leaf spot, red mite damage, green mite damage, cassava brown streak disease, and cassava mosaic disease all had 98 percent, 96 percent, 95 percent, and 95 percent recognition rates, respectively.

Inception and GoogleNet GoogLeNet [39] (Szegedy et al., 2014) is a CNN structure with an inception module. This study [40] (Szegedy, et al., 2014) looked at the performance of neural networks, specifically the impact on computation speed caused by increased network size and kernel size homogeneity, which is prone to inefficiency when dealing with features of diverse forms. GoogLeNet presents a unique neural network design that takes advantage of sparsity. Previous research has suggested that clustering sparse matrices can significantly improve performance. GoogLeNet presents inception modules that take advantage of a convolutional vision network's local sparse structure. The idea behind this notion is that the filters should be able to capture large-scale data while still retaining fine-resolution information.

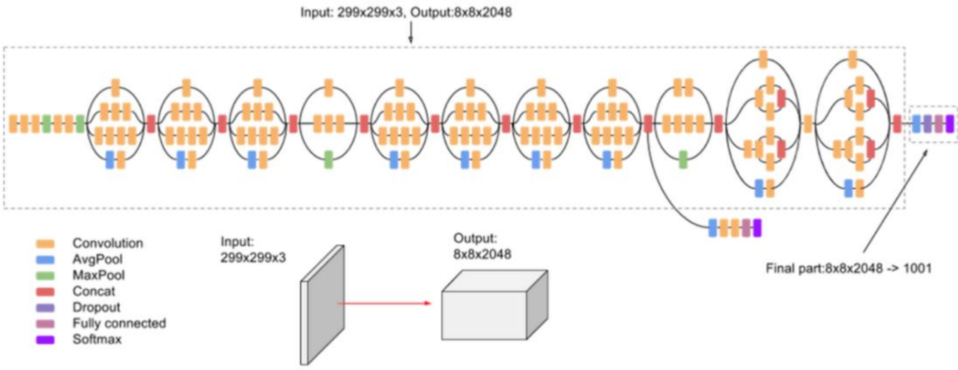

Figure 4: Inception V3 architecture

2.2.2.1 Configuration details of InceptionV3

Convolutional Layers	48
Dropout	0.5
Learning Rate	0.0001
Activation Function	ReLU
Batch Size	32
Epochs	20
Training Algorithm	Adam

Table 2: InceptionV3 Configuration

2.2.3 MobileNetV2 Architecture

MobileNet is a convolutional neural network that is used for classification, detection, and other common tasks. They are relatively compact, which allows us to use them on mobile devices, and their size is 17MB. [37] is the architect behind this structure. A streamlined architecture is used to create these. This design built lightweight and deep neural networks using depth-wise separable convolutions. These depth-wise convolutions provide a factorization effect, reducing the model's size and speeding up processing. Mobile Nets can be used to improve efficiency in a variety of applications. MobileNets can be utilised for a variety of applications, including object identification, fine grain classification, face attribute categorization, and so on. MobileNet is a powerful convolutional neural network that can be utilised in computer vision.

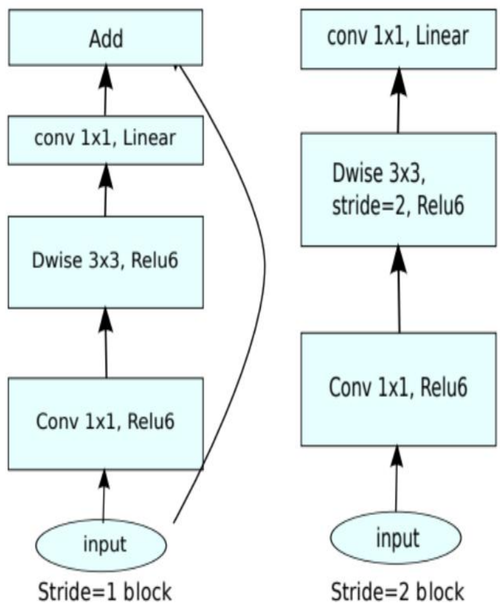

Figure 5: MobileNetV2 Architecture

2.2.3.1 Configuration details of MobileNetV2

Convolutional Layers	3 (with 3x3 and 1x1)
Dropout	0.3
Learning Rate	0.0001
Activation Function	ReLU
Batch Size	32
Epochs	20
Training Algorithm	Adam

Table 3: MobileNetV2 Configuration

With a self-made dataset of 5000 photos of various resolutions, Alvaro Fuentes, Sook Yoon, Sang Cheol Kim, and Dong Sun Park [41] developed a robust DL-based detection of tomato plant disease in real time and pest recognition. To determine which of three types of detectors is best for the task, they employed Faster R-CNN, R-FCN, and SSD. As deep feature extractors, they used VGG net and ResNET (Residual Network). They were able to recognise 9 different types of diseases and pests with an average AP of 83 percent, indicating that Faster R-CNN in combination with VGG-16 produced the greatest results.

On a Plant Village dataset of 50,000 photos, Guan Wang, Yu Sun, and Jianxin Wang [42] employed VGG-16, VGG-19, Inception-v3, and ResNet50 to automatically detect the harmful consequences of a plant illness. With a spread of 38 class labels, they were able to attain an accuracy of 80.0 to 90.4 percent.

To train a deep CNN, Yu-Dong Zhang et al. [43] used a hybrid dataset (3,600 images overall) that included both self-taken and internet-acquired photos. They used a 14-layer deep CNN with data augmentation to categorise 18 different varieties of fruits, with an overall max-pooling accuracy of 94.94 percent. Gamma correction, picture rotation, and noise injection were employed as data augmentation techniques.

Israr Hussain, Qianhua He, and Zhuliang Chen [44] trained a DCNN model to automate fruit recognition, primarily for commercial source trace systems, using a self-made dataset. There were 44,406 photos in the dataset, with 15 different fruit categories and sub-categories, such as six different types of apple fruits. They claimed to be able to attain a 99 percent accuracy rate and to “effectively meet real-world application requirements.”

Deep learning was applied by Dor Oppenheim, Guy Shani, Orly Erlich, and Leah Tsror [45] to detect and identify potato tuber disease. The 2,465 photos in the self-created dataset were split into four disease categories and a fifth category of healthy potatoes. After fine-tuning the algorithms and the requisite image collection, they were able to attain an accuracy of up to 96 percent on the quicker R-CNN.

In melon plants, Mónica Pineda, Mara L. Pérez-Bueno, and Matilde Barón [46] were able to detect bacterial infection. They employed classification algorithms to divide imaging data from a self-administered dataset of 2,465 pictures into five unique types. To categorise the photos, they utilised three algorithms: logistic regression analysis (LRA), support vector machine (SVM), and artificial neural network (ANN). They used the same dataset to compare the performance of the three algorithms. The three performed differently depending on the plant dosage, with a range of 96 to 99 percent variance.

Edna Chebet is a well-known author. A comparison of employing fine-tuned DL models for the diagnosis of plant diseases was also published by Li Yujiana, Sam Njukia, and Liu Yingchun [47]. VGG16, Inception V4, ResNet with various layers, and DenseNet with 121 layers were utilised in the comparison, with Keras and Theano being used to train the architectures. Plant Village provided the dataset, which included 54,306 photos separated into 38 classes. They concluded that DenseNets performed the best and that its accuracy increased as the number of epochs increased, eventually reaching 99.75 percent accuracy.

Horea Murşean and Mihai Oltean [48] used a new plant dataset from Fruit 360 in 2018. We, too, used this resource as one of our dataset sources in our research. Horea et al. trained a CNN to recognise fruits from photographs using a dataset of 69,905 photos. They had a 94.16 percent total accuracy rate.

Aydin Kayaa et al. [49] investigated transfer learning for a plan classification model based on deep neural networks. The strategy attempts to complement pre-existing expertise of distinct DNN applications, which had previously been employed in isolation, and merge it with newer technology. For this investigation, four separate datasets were used, all of which were freely available. Flavia and Swedish Leaf were the first two datasets, followed by the UCI Leaf dataset and the Plant Village dataset. The dataset contained almost 54,000 photos for the training of the various architectures. The plant classification was done using a total of 5 DNN. The first approach was self-developed, followed by fine-tuning the CNN, cross-dataset fine tuning, the use of pre-trained models, and finally, a mix of recurrent neural network and CNN algorithms (AlexNet and VGG16). When trained on the Plant Village dataset, the AlexNet/VGG-16 models had the best results, with an accuracy of 99.80%.

Peng Jiang, Yuehan Chen, Bin Liu, Dongjian He, and Chunquan Liang [50] proposed employing deep learning and upgraded CNNs to identify apple-leaf illnesses in real time. They trained an SSD with Inception module and Rainbow concatenation using a self-made dataset of 26,377 photos divided into five classes (INAR-SSD). They were able to obtain a 78.8% accuracy rate.

This study [51] based on diagnosis of tomato disease by using the deep convolutional neural networks and object detection models which were Faster R-CNN and Mask R-CNN. For the identification of the disease Faster R-CNN is used for the detection and segmentation of infected area and shape Mask R-CNN has been used.

In early 90's researcher adopted technology for the identification and diagnostic purpose of plant disease. In [52] genetic algorithms is being used for the measurement of characteristics like form and spectrum reflection to classify the identification factors. Conversely, the calculated identification factors was not up to the mark as expected due to the deficiency of the material like colour and texture information.

2.3 Convolutional Neural Networks (CNN)

Deep learning, according to Li Deng and Dong Yu in [53], is a type of machine learning algorithms that employs numerous layers to extract features from raw input. The retrieved features are at a higher level, allowing for more detail in the model. Rina Dechter introduced deep learning to machine learning in 1986 [54]. Since then, a number of advancements have been made, including the construction of neural networks that can work in both supervised and unsupervised environments. Deep learning, often known as convolutional neural networks (CNN), is a type of unsupervised machine learning.

Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, who developed the LeNet [55], were among the forerunners in the field. It was a neural network with seven levels that was used to classify handwritten digits on checks. Its main stumbling block was the huge computational demand. The CNN was created by Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton in 2012. ImageNet's annual ImageNet Large Scale Visual Recognition Challenge (ILSVRC) was won by AlexNet [56]. The network had a similar architecture to LeNet, although it was deeper and had more filters. 11x11, 5x5, 3x3, convolutions, max pooling, dropout, data augmentation, ReLU activations, and SGD with momentum were among the filters used. Then, in 2013, ZFNet, another CNN built by Zeiler, Matthew D., and Rob Fergus [57], won the ILSVRC, with a top-5 error rate of 14.8 percent, better than AlexNet, which had a top-5 error rate of 15.3 percent the year before. This was accomplished mostly through fine-tuning and adjusting the AlexNet architecture's hyper-parameters. GoogleNet, called Inception V1 [62] in the journal, won the competition in 2014. With a top-5 error rate of 6.67 percent, it became the first great leap following AlexNet. It used batch normalisation, picture distortions, and RMSprop and was based on the LeNet architecture. The Inception Module is the name given to this new invention. The number of parameters was reduced by utilising a 22-layer deep CNN to lower the parameter from 60 million to 4 million in AlexNet. The VGGNet, built by the Oxford Robotics Institute's Simonyan, Karen, and Zisserman, Andrew [58], came in second to the GoogleNet.

The VGGNet had 16 layers of 3x3 convolutional convolutions and more filters than the AlexNet. Its consistent structure makes it a go-to feature extractor for a variety of applications. Then, in 2015, ResNet, formally known as the Residual Neural Network (RNN) [64], won the ILSVRC title. RNN uses skip connections, also known as gated units, to allow it to utilise 152 layers while maintaining a lower computational complexity than VGGNet. It had a 3.57 percent top-five mistake rate.

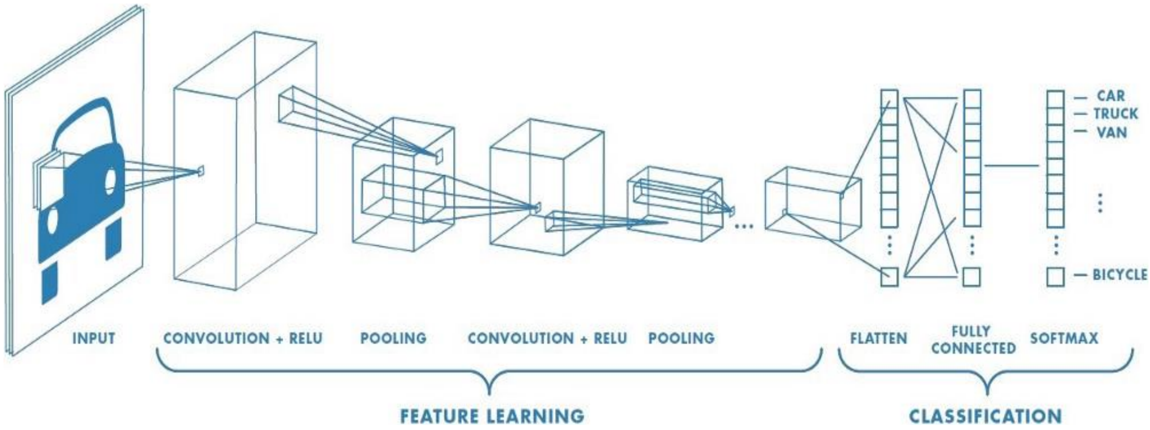

Figure 6 Convolutional Neural Network

2.3.1 Convolution Layers

Convolutional Layers are in charge of convolution the input image with the filter in order to extract the required features and generating a feature map based on the filter size. The size of the Input Image determines the filter size. The filter is made up of two parts: a filter size F and a total number of filters K. The Input Image Dimensions (W(i) * H(i) * D(i)) would be the convolutional Layer's input, and the output would be (W(o) * H(o) * D(o)), where D(o) is equal to the total number of filters K, and W(o) and H(o) may be computed using the following equation [59].

$$\frac{((W(i),\ H(i)) - F) + 2p}{Stride + 1}, \qquad (3)$$

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

2.3.2 Parameter Calculation

To acquire the overall trainable and non-trainable parameters in the model, as well as the network's total memory usage, parameters for each convolution layer are calculated. If we have an input of (W(i) * H(i) * D(i)) and a convolution filter of (W(f) * H(f) * D(f)), where W(i), H(i), and D(i) are the width, height, and total number of feature maps in a convolution filter and W(f), H(f), and D(f) are the width, height, and total number of feature maps in a convolution filter, then As a result, the parameters can be computed using the formula below.

$$(W(f) * H(f) * D(i) + 1) * D(f)\,, \qquad (4)$$

Equation 2 Parameter Calculation of Layers

2.3.3 Pooling Layers

Pooling Layers are utilised to reduce the total number of parameters used in the network while also lowering the overall computational cost [60]. Average Pooling and Max Pooling are two of the most often utilised pooling strategies.

2.3.4 Dropout Layers

Dropout Layers were created to prevent the model from overfitting or underfitting the dataset. It determines the number of nodes to be employed in the training process. After completely connected layers that are prone to overfitting, these layers are typically utilised [61].

2.3.5 Activation Function

The activation function aids in establishing a non-linear relationship between the image class and Image Data. They determine whether a neuron should be triggered or not based on the neuron's relevance to the desired output [62]. Various activation functions, such as tanh, sigmoid, Relu, Leaky Relu, and others, are used.

2.3.6 Optimization Techniques

The weights for your model are calculated using optimization approaches. They adjust the weights during the learning process until you achieve the desired result. SGD, SGD with momentum, NAG, Adagrad, RMSprop, and Adam [63] are among the optimization algorithms employed.

2.3.7 Flatten Layers

Flatten Layers are in charge of turning data into a one-dimensional vector that may then be passed into Fully Connected Layers for classification.

2.3.8 Fully Connected Layers

The feed forward neural networks are what they're called. To compute the classification, the first FC layer takes the data from the last convolution layer after it has been flattened into a one-dimensional vector, and the last FC layer delivers the final probabilities determined for each label.

2.3.9Accuracy Calculation

The f1 Score [64], which contains two metrics Precision and Recall, is used to calculate it. Precision refers to the number of true class predictions that are true to the real class, whereas recall refers to the total number of true class predictions made over all true samples in the dataset.

The formulas for each are listed below.

$$Precision = \frac{TP}{TP + FP}, \qquad (4)$$

Equation 3 Precision

$$Recall = \frac{TP}{TP + FN}, \qquad (5)$$

Equation 4 Recall

$$f1\ Score = 2\ .\frac{Precision\ .\ Recall}{Precission\ +\ Recall}, \qquad (6)$$

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 approach, we begin by gathering datasets from available resources, and then process them according to the needs, which include image scaling, data augmentation, combining two datasets, and class labelling. The data is then split between the Train and Validation sets and put into our proposed CNN. Afterwards, the trained model is tested on individual photos to ensure that it is performing as expected.

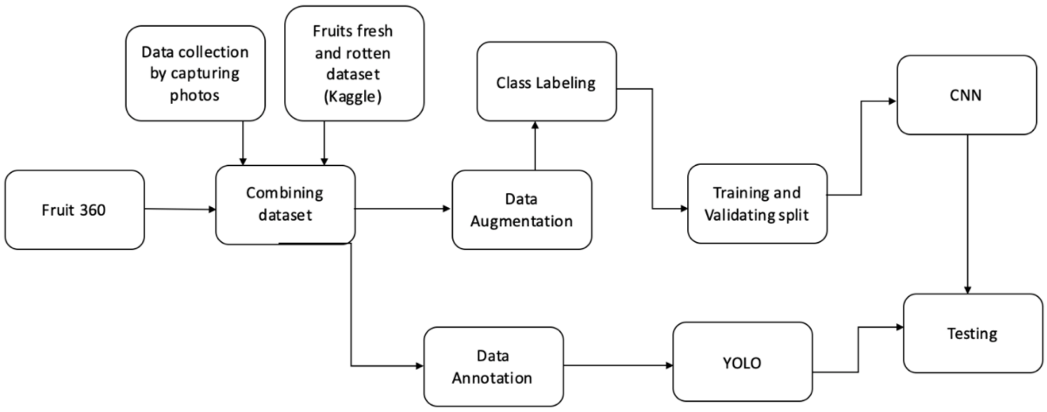

Figure 7 Proposed Scheme Diagram

3.2 Implementation

3.2.1 Data Collection

a. Fruits Fresh and Rotten for Classification (KAGGLE)

We have used all of the 6 classes of this dataset which contains 13.6k images.

b. Fruit 360 Dataset by Horea [7] (Fruits and Vegetables)

We have used two classes of this dataset (Fresh Pomegranate and Fresh Tomato) which contains 2001 images.

c. Images captured through Mobile Phone

We have added two classes (Rotten Pomegranate and Rotten Tomato) by capturing through mobile which contains 1972 images

3.2.2 Dataset Samples

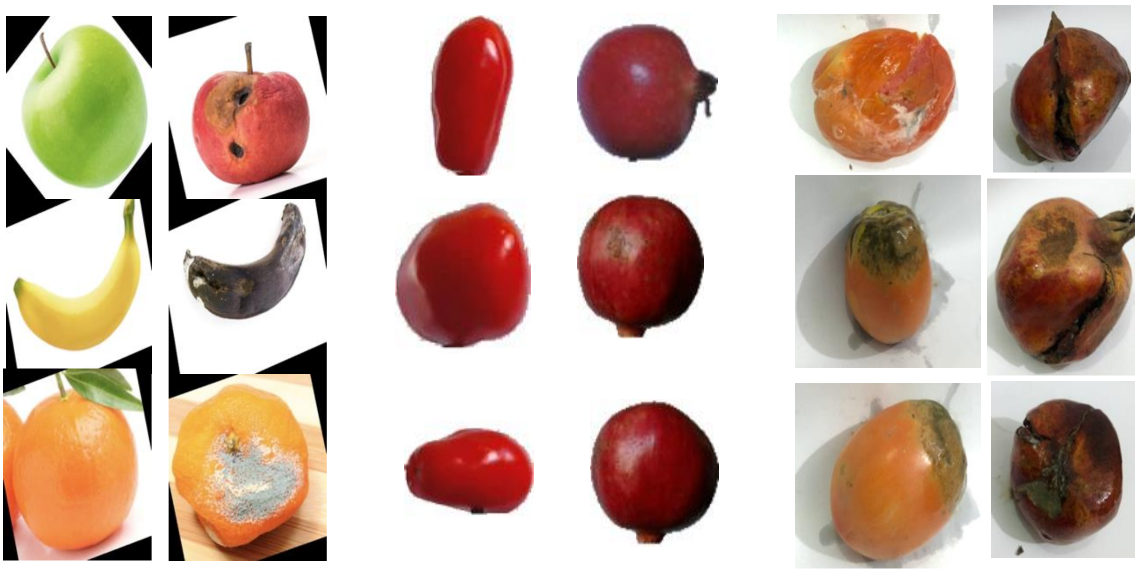

Figure 8: Fresh and Rotten Fruits dataset

3.3 Dataset Distribution

- Complete Dataset= Total 10 classes with 17930 images.
- Dataset is divided into three parts Training Data, Validation Data and Test Images.
- Training and Validation is done in split of (80,20) respectively.
- Test Images were separated to test the networks on singular inputs and a batch of 128 images.

3.4 Data Augmentation

We have augmented the data for the following reasons:

- To balance the imbalance classes for avoiding overfitting
- To add more data with variation.

I have used the following techniques to augment the data:

- Width Shift
- Height Shift
- Brightness
- Horizontal Flip

3.5 Data Annotation

- We have used online source makesense.ai to annotate our dataset

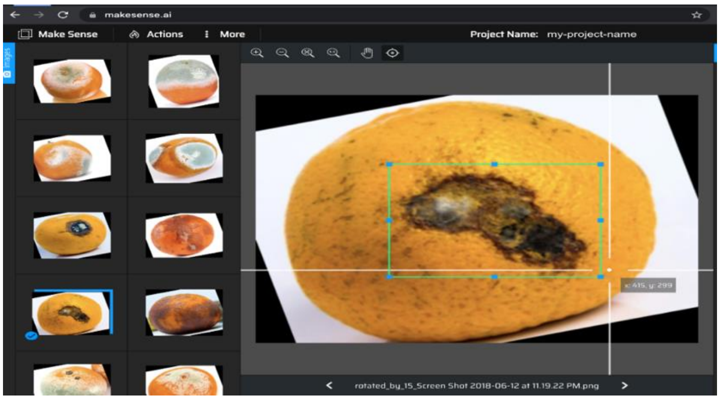

Figure 9: Data Annotation

4 Results and Discussion

4.1 Results of VGG16:

This section discusses the results of the VGG16 trained network on the given dataset for the Batch Size (BS) of 32 with the data split of (80, 20).

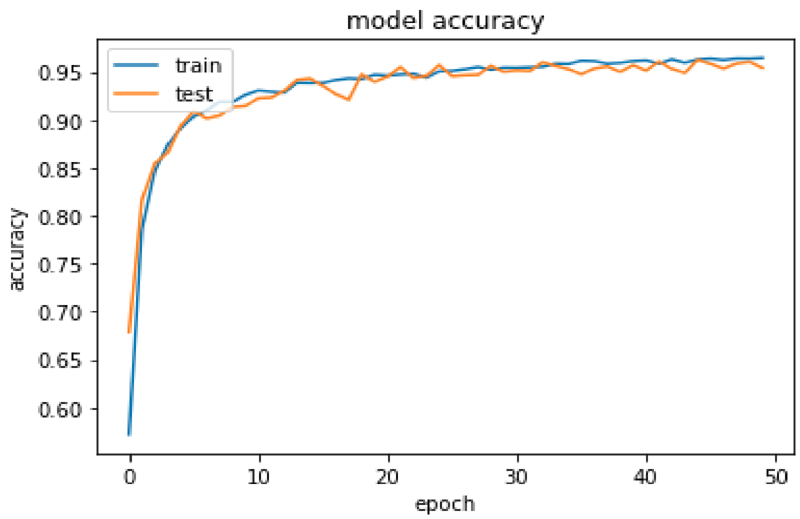

Figure 10 Model Accuracy for VGG16

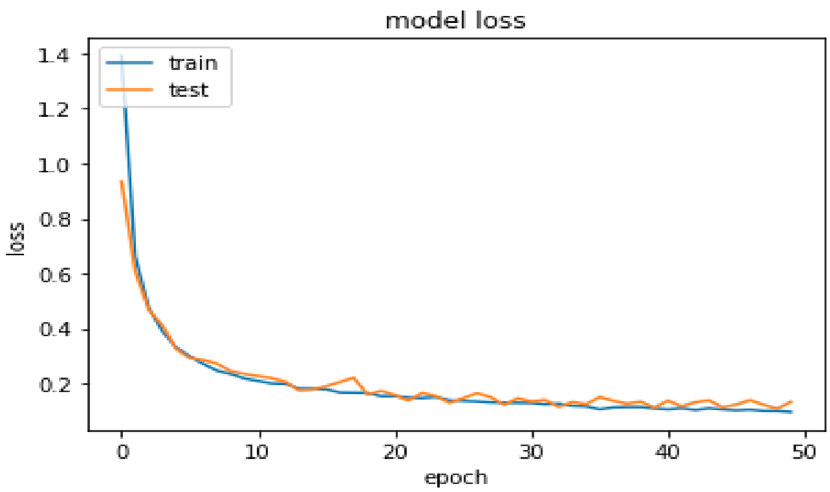

Figure 11 Model Loss for VGG16

According to Figure (10) and (11) it is clearly seen that the training accuracy and the validation accuracy for VGG16 model on BS of 32 over DS of (80, 20) is 96.23 and 96.24 percent respectively. The training loss and the validation loss for VGG16 model is 10.65 and 10.55 percent respectively.

4.2 Results of InceptionV3:

This section discusses the results of the VGG16 trained network on the given dataset for the Batch Size (BS) of 32 with the data split of (80, 20).

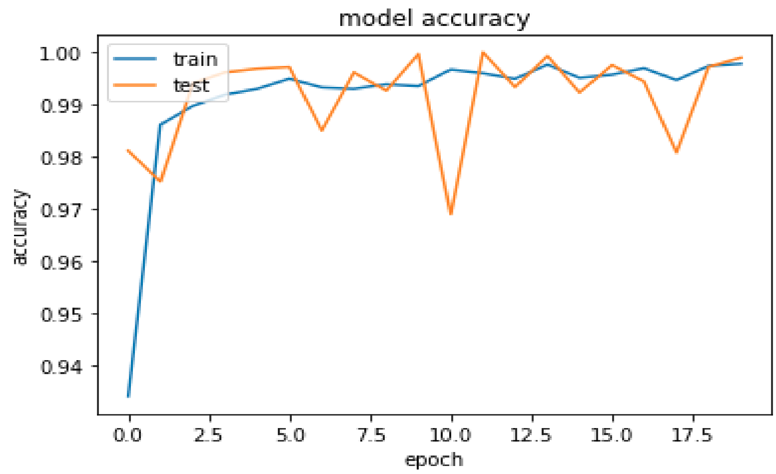

Figure 12 Model Accuracy for InceptionV3

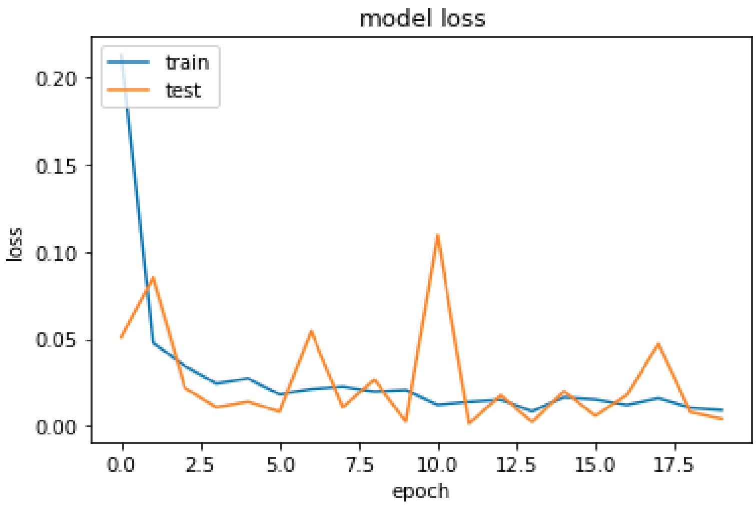

Figure 13 Model Loss for InceptionV3

According to Figure (12) and (13) it is clearly seen that the training accuracy and the validation accuracy for InceptionV3 model on BS of 32 over DS of (80, 20) is 99.94 and 99.90 percent respectively. The training loss and the validation loss for InceptionV3 model is 0.25 and 0.37 percent respectively.

4.3 Results of MobileNetV2:

This section discusses the results of the MobileNetV2 trained network on the given dataset for the Batch Size (BS) of 32 with the data split of (80, 20).

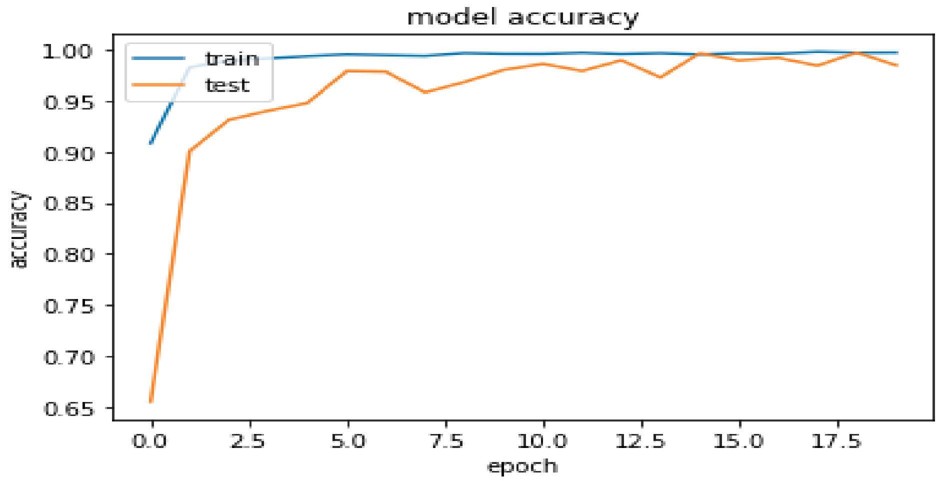

Figure 14 Model Accuracy for MobileNetV2

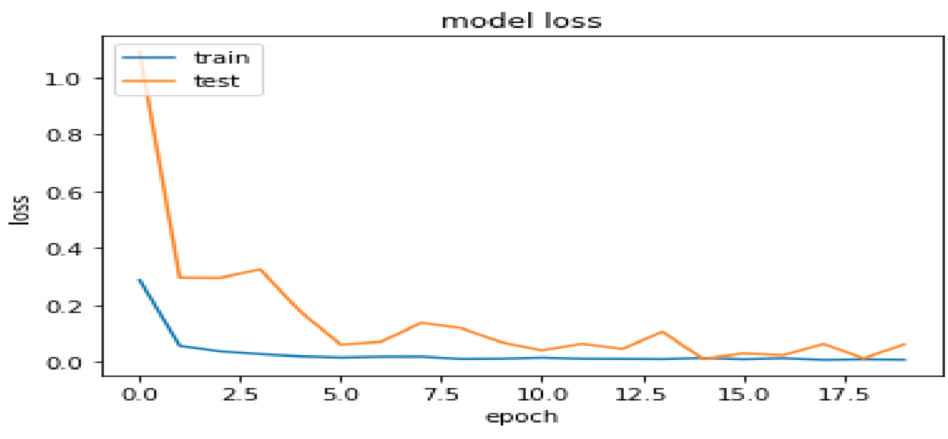

Figure 15 Model Loss for MobileNetV2

According to Figure (14) and (15) it is clearly seen that the training accuracy and the validation accuracy for MobileNetV2 model on BS of 32 over DS of (80, 20) is 99.87 and 98.43 percent respectively. The training loss and the validation loss for InceptionV3 model is 0.47 and 5.42 percent respectively.

4.4 Results Comparison

According to the above-mentioned results, the comparison table of the different models' accuracy and loss is given below:

Model	Training Accuracy (%)	Validation Accuracy (%)	Training Loss (%)	Validation Loss (%)
VGG16	96.23	96.24	10.65	10.55
InceptionV3	99.94	99.90	0.25	0.37
MobileNetV2	99.87	98.43	0.47	5.42

Table 4 Results Comparison

4.5 Classification Reports

4.5.1 VGG16

```
              precision    recall  f1-score   support

         0.0       1.00      1.00      1.00        11
         1.0       0.94      1.00      0.97        16
         2.0       0.94      0.94      0.94        16
         3.0       1.00      1.00      1.00         5
         4.0       1.00      1.00      1.00        14
         5.0       1.00      0.95      0.98        22
         6.0       1.00      1.00      1.00        23
         7.0       0.91      0.91      0.91        11
         8.0       1.00      1.00      1.00         4
         9.0       1.00      1.00      1.00         6

    accuracy                           0.98       128
   macro avg       0.98      0.98      0.98       128
weighted avg       0.98      0.98      0.98       128
```

Figure 16 Classification Report of VGG16

4.5.2 InceptionV3

	precision	recall	f1-score	support
0.0	1.00	1.00	1.00	2
1.0	1.00	1.00	1.00	3
2.0	1.00	1.00	1.00	6
3.0	1.00	1.00	1.00	2
4.0	1.00	1.00	1.00	2
5.0	1.00	1.00	1.00	5
6.0	1.00	1.00	1.00	3
7.0	1.00	1.00	1.00	7
8.0	1.00	1.00	1.00	1
9.0	1.00	1.00	1.00	1
accuracy			1.00	32
macro avg	1.00	1.00	1.00	32
weighted avg	1.00	1.00	1.00	32

Figure 17 classification Report of InceptionV3

4.5.3 MobileNetV2

	precision	recall	f1-score	support
0.0	1.00	1.00	1.00	5
1.0	1.00	1.00	1.00	1
2.0	1.00	1.00	1.00	3
3.0	1.00	1.00	1.00	3
4.0	1.00	1.00	1.00	3
5.0	1.00	1.00	1.00	10
6.0	1.00	1.00	1.00	3
7.0	1.00	1.00	1.00	2
9.0	1.00	1.00	1.00	2
accuracy			1.00	32
macro avg	1.00	1.00	1.00	32
weighted avg	1.00	1.00	1.00	32

Figure 18 Classification Report for MobileNetV2

4.6 Classification Test Results

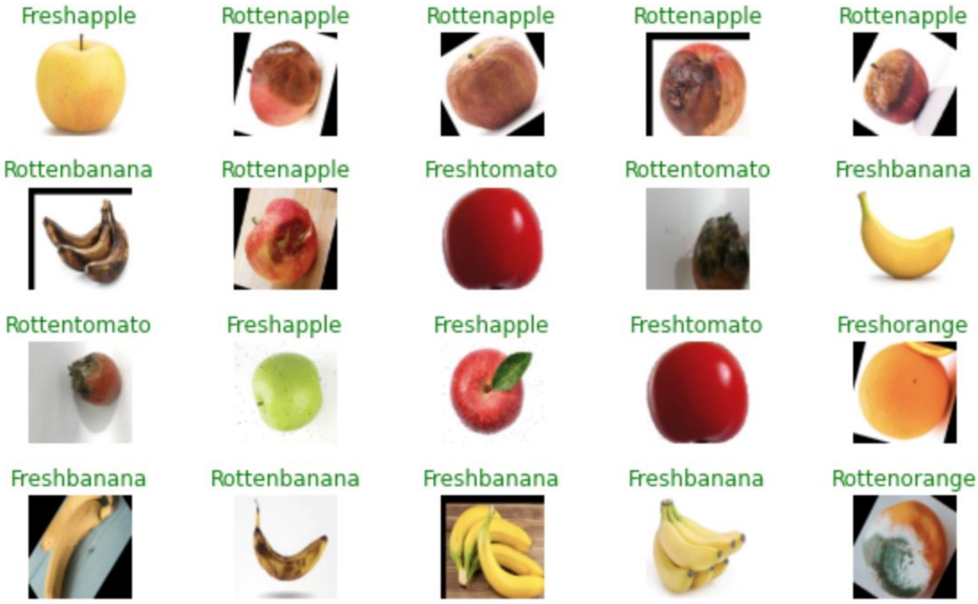

Figure 19 Test Results

Figure (19) represents the test results over the batch of 128 images. Green Color represents that the model has classified the images correctly. Red color represents that the model has classified the images incorrectly.

4.7 Results of YOLOV5

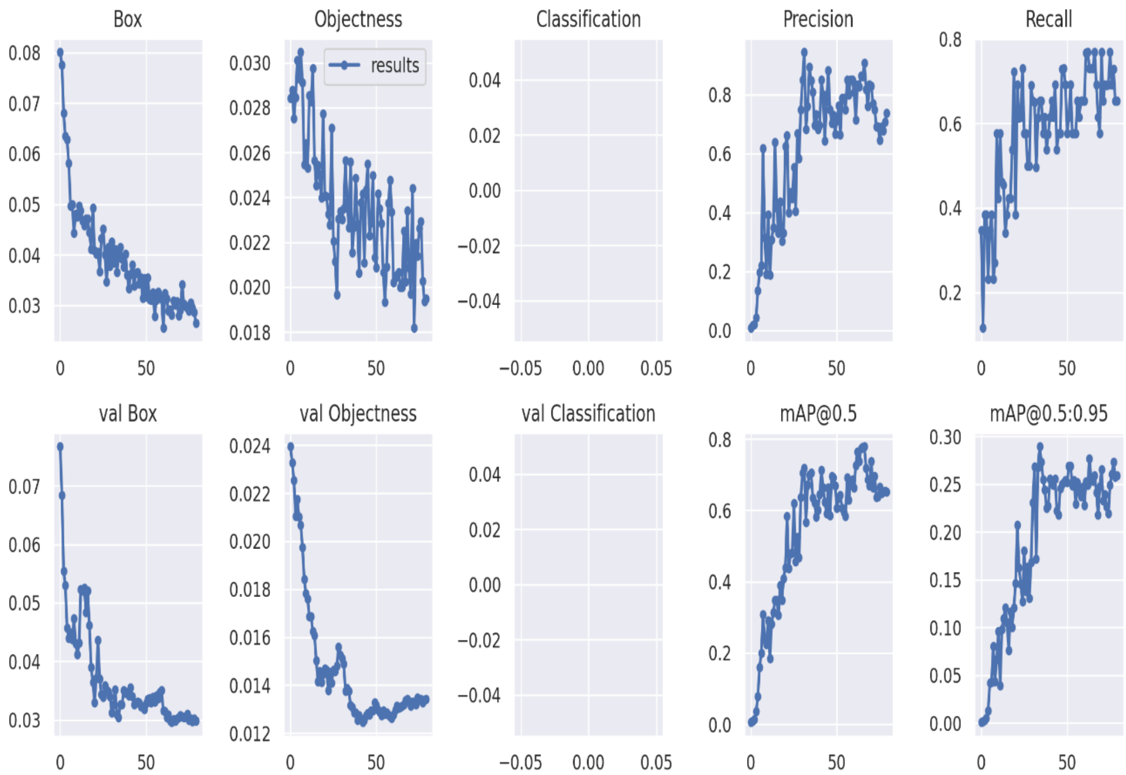

Figure 20 YOLOV5 Results

4.8 Results of infection part detection

Figure 21 Test Results of YOLOV5

5 Future Work

- We can use:
    - Other Data Augmentation Techniques and networks to achieve lower loss
    - High Resolution to achieve more detailed results
    - Faster Mask RCNN for the detection of the infected area
- 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 applied *Transfer Learning* using three pre-trained models for the classification of fresh and rotten fruits. Dataset is collected through two online resources and we have added two classes by capturing images through mobile phone. InceptionV3 provides best accuracy 99.90 on our dataset.
We have also annotate the images to detect the rotten part of the fruit. For detection of the rotten part, we have applied the YOLOV5 model and achieved good results.

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, BatchNormalizati
on, Flatten, Conv2D
from keras.layers import AveragePooling2D, MaxPooling2D, Dropout, GlobalMaxPo
oling2D, 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
, BatchNormalization
#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, BatchNorm
alization, Flatten, Conv2D, AveragePooling2D, MaxPooling2D, GlobalMaxPooling2
D
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 = EPOC
HS)
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)
_fv1_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')
```

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] S. Péneau, A. Linke, F. Escher, and J. Nuessli, "Freshness of fruits and vegetables: consumer language and perception," *British Food Journal*, vol. 111, no. 3, pp. 243–256, 2009.

[8] K. Sindhi, J. Pandya, and S. Vegad, "Quality evaluation of apple fruit: A Survey," *International Journal of Computer Applications*, vol. 136, no. 1, pp. 32–36, 2016.

[9] A. K. Shukla, "Preface," *Electron Spin Resonance in Food Science*, pp. xiii-xiv, 2017.

[10] J. B. Cunha, “Application of image processing techniques in the characterization of plant leafs,” *2003 IEEE International Symposium on Industrial Electronics (Cat. No.03TH8692).*

[11] Y. Koumpouros, B. D. Mahaman, M. Maliappis, H. C. Passam, A. B. Sideridis, and V. Zorkadis, “Image processing for distance diagnosis in pest management,” *Computers and Electronics in Agriculture*, vol. 44, no. 2, pp. 121–131, 2004.

[12] T. Brosnan and D.-W. Sun, “Inspection and grading of agricultural and food products by computer vision systems—a review,” *Computers and Electronics in Agriculture*, vol. 36, no. 2-3, pp. 193–213, 2002.

[13] E. Ozyildiz, N. Krahnstöver, and R. Sharma, “Adaptive texture and color segmentation for tracking moving objects,” *Pattern Recognition*, vol. 35, no. 10, pp. 2013–2029, 2002.

[14] D. M. Barrett, J. C. Beaulieu, and R. Shewfelt, “Color, Flavor, Texture, and Nutritional Quality of Fresh-Cut Fruits and Vegetables: Desirable Levels, Instrumental and Sensory Measurement, and the Effects of Processing,” *Critical Reviews in Food Science and Nutrition*, vol. 50, no. 5, pp. 369–389, 2010.

[15] J. Lequeu, M.-L. Fauconnier, A. Chammaï, R. Bronner, and E. Blée, “Formation of plant cuticle: evidence for the occurrence of the peroxygenase pathway,” *The Plant Journal*, vol. 36, no. 2, pp. 155–164, 2003.

[16] D. M. Barrett, J. C. Beaulieu, and R. Shewfelt, “Color, Flavor, Texture, and Nutritional Quality of Fresh-Cut Fruits and Vegetables: Desirable Levels, Instrumental and Sensory Measurement, and the Effects of Processing,” *Critical Reviews in Food Science and Nutrition*, vol. 50, no. 5, pp. 369–389, 2010.

[17] K. Bresilla, G. D. Perulli, A. Boini, B. Morandi, L. Corelli Grappadelli, and L. Manfrini, “Single-Shot Convolution Neural Networks for Real-Time Fruit Detection Within the Tree,” *Frontiers in Plant Science*, vol. 10, 2019.

[18] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You Only Look Once: Unified, Real-Time Object Detection,” *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* 2016.

[19] G. Zeng, “Fruit and vegetables classification system using image saliency and convolutional neural network,” *2017 IEEE 3rd Information Technology and Mechatronics Engineering Conference (ITOEC),* 2017.

[20] J. Gómez-Sanchis, L. Gómez-Chova, N. Aleixos, G. Camps-Valls, C. Montesinos-Herrero, E. Moltó, and J. Blasco, “Hyperspectral system for early detection of rottenness caused by Penicillium digitatum in mandarins,” *Journal of Food Engineering*, vol. 89, no. 1, pp. 80–86, 2008.

[21] C. V. Reddy, U. S. Reddy, and K. V. Kishore, “Facial Emotion Recognition Using NLPCA and SVM,” *Traitement du Signal*, vol. 36, no. 1, pp. 13–22, 2019.

[22] K. Thenmozhi and U. Srinivasulu Reddy, “Crop pest classification based on deep convolutional neural network and transfer learning,” *Computers and Electronics in Agriculture,* vol. 164, p. 104906, 2019.

[23] C. V. Ramireddy and K. V. Kishore, “Facial expression classification using Kernel based PCA with fused DCT and GWT features,” *2013 IEEE International Conference on Computational Intelligence and Computing Research,* 2013.

[24] C. V. Reddy, K. V. K. Kishore, U. S. Reddy, and M. Suneetha, “Person identification system using feature level fusion of multi-biometrics,” *2016 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC),* 2016.

[25] V. Chirra, S. ReddyUyyala, and V. KishoreKolli, “Deep CNN: A Machine Learning Approach for Driver Drowsiness Detection Based on Eye State,” *Revue d'Intelligence Artificielle*, vol. 33, no. 6, pp. 461–466, 2019.

[26] 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.

[27] ouhan, 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.

[28] al, 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.

[29] 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/.

[30] 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.

[31] K. Bresilla, G. D. Perulli, A. Boini, B. Morandi, L. Corelli Grappadelli, and L. Manfrini, “Single-Shot Convolution Neural Networks for Real-Time Fruit Detection Within the Tree,” *Frontiers in Plant Science,* vol. 10, 2019.

[32] 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.

[33] 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.

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

[35] 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.

[36] 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.

[37] 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.

[38] 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.

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

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

[41] 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.

[42] 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.

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

[44] 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.

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

[46] 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.

[47] 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].

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

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

[51] 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.

[52] 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].

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

[54] 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.

[55] 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.

[56] 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.

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

[58] 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.

[59] 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.

[60] 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.

[61] 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.

[62] 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.

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

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