Multi-Class Image Classification of Fruits and Vegetables Using Transfer Learning Techniques

DISSERTATION SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MSc Data Analytics

Author:
Hanshu Tomar *(10505823)*

Supervisor:
Prof. Basel Magableh

May 20, 2020

DECLARATION:

'I declare that this dissertation that I have submitted to Dublin Business School for the award of MSc in Data Analytics is the result of my own investigations, except where otherwise stated, where it is clearly acknowledged by references. Furthermore, this work has not been submitted for any other degree.'

Signed: Hanshu Tomar
Student Number: 10505823
Date: May 2020

Acknowledgement:

It is a privilege for me to thank every single person who contributed to this study and made it possible. First of all, I would like to express my sincere thanks to my supervisor, Prof. Basel Megableh, for supporting and guiding me by sharing his objective, giving me the significant direction from the start of the study to the last stage.

I would also like to thank all other individuals, my teachers and fellow classmates who helped me dive deep into the subject, understand the concepts directly and indirectly. Finally, I would also like to thank Dublin Business School for providing the opportunity to be part of this masters program and performing this study.

Abstract

The need for an efficient fruit and vegetable identification and classification is always important and beneficial for not only the agricultural department or food processing industry, but also to the low level retail stores and supermarkets where fruits and vegetables are sold. Building an efficient automated tool is very much required. In order to build this application, an efficient and effective classification model has to used, which can classify 1000's of fruits and vegetables in seconds. The purpose of this study is to find the best classifier model which can be used to build this automated application. While many advancements have been made in recent years, many methods still struggle from prolonged training and testing time and even significantly more number of false positives after classification. Thereby, in this paper, a review and experiments on 7 different available transfer learning models such as VGG16, ResNet50, MobileNet, DenseNet, InceptionV3, xception and InceptionResNet is conducted and compared by there accuracy, precision, F1 Score and training time, so that an effective and efficient automated classifying system can be built in future. Along with this, a self-designed CNN model is trained and tested. The experiments are conducted using Fruit 360 dataset of 120 classes. Initial phase of this study involves training the models using subset of the dataset with 21 classes. VGG16 and ResNet50 are resulted as top 2 models. Thereby, the later phase of the experiment is conducted on these two models on the whole dataset with 120 classes. The overall results show VGG16 is the best model with 99% of training accuracy and 95% of testing accuracy.

Keywords:
Multi-Class Image Classification; Fruit Classification; Transfer Learning; Convolutional Nueral Network; VGG16; ResNet50.

Contents

List of Figures

1 Introduction

Food is the basic need of a human being and the main substances are fruits and vegetables. As per the scientists there are "400,000 plant species available on the earth" out of which humans can eat 300,000 specie types. However, we only consume "200 species" globally [Barnett, 2015]. Out of these many edible species there are 2000 different types of fruits found all over the world [Fruitsinfo.com, 2020] and in order to differentiate these types it becomes important to build an automated fruit or a vegetable classifier. An efficient classifier is not only important for the agricultural department or for a food processing industry it is also very important at the lower retail store or at a supermarket where the whole process of billing can be eased just by placing the selected fruit in front of the camera built on the weighing scale system which can give the price accordingly.

Fruit classification based on its type was majorly built using image processing and computer vision as explained by researchers in articles [Zhang et al., 2014, Adigun et al., , Zhang et al., 2014], the drawback of using such techniques is that data/images should be pre-processed first and then the classifier is trained. In recent years Machine Learning techniques has enhanced various applications in Artificial Intelligence domain. Using ML algorithms many advancements have been done in the food processing and agricultural departments [Kamilaris and Prenafeta-Boldú, 2018]. Like wise, many new techniques were used in the same field with the aid of deep learning [Larada et al., 2018, Mureşan and Oltean, 2018]. Nevertheless, all these methods take a long time to train, test and provide minimal accuracy, which can hinder real-time use. In consideration of the circumstances, a appropriate pattern for fruit classification must be sought.

Neural networks are one of the most impressive Machine Learning models in order to learn from labeled data and then treat the unlabeled data [Svozil et al., 1997]. It is necessary for neural networks to have sufficient number of training data to give better results. But, it is not necessary to fulfill such a conditions in real-time, because labelled data can only be collected by a manual procedure which is both time-consuming and error-prone [Kamilaris and Prenafeta-Boldú, 2018]. To end this difficulties, more stable practice of training the deep learning models also known as fine-tuning is applied on the large dataset which is called as "Transfer Learning" is used [Huang et al., 2019]. Transfer learning enables current parameters (convolution weights) to be re-used from a trained model on large amount of data to train new models from significantly limited training data [Koirala et al., 2019]. There are many such freely available public datasets exists which enables users to use the labelled data of basic things/objects, most widely used ones are ImageNet, PASCAL VOC and COCO.

The theory of Transfer Learning is inspired by the idea that previously acquired experience can be intelligently implemented to overcome new issues more easily or with better alternatives. In an NIPS-95 workshop on "Learning to Learn," [to Learn, 1995] which concentrated on reused previously acquired knowledge, the underlying impetus for Transfer learning in the field of machine learning was addressed. Transfer Learning is a process where the weights from the network layers of existing model, which is called as pre-trained model is used or replicated to a new model. These weights are the parameters which holds feature values such as shape, colour, size, etc. of the basic object images. In order to achieve classification on the new dataset, the last classification layer of the pre-trained model has to be replaced with the new layer with respect to the new dataset and desired number of classes. Transfer Learning will not give better results if the target classification image object does not belong to common things which are pre-trained. There are few

instances where transfer learning holds no good approach, for example: Results from the paper [Bargoti and Underwood, 2017b] show no difference in accuracy or performance, in order to identify apple images by Faster-RCNN model using the pre-trained weights from ImageNet and other orchards. However, as our experiments are on the fruit and vegetable images, Transfer Learning using ImageNet pre-trained weights is the best fit as it contains several images of fruits and vegetables.

The motive of this research is to review and examine the efficiency of Transfer Learning and Fine Tuning multi class image classification accuracy. Fruits and vegetables classification has always been a difficult task as many fruits are very similar to each other or sometimes the same kind of fruit is different in shape, size and even in colour. While many advancements have been made in recent years, many methods still struggle from prolonged training and testing time and even significantly more number of false positives after classification. Thereby, in this paper, a review on different available transfer learning models is conducted and compared by there accuracy, precision, F1 Score and training time, so that an effective and efficient automated classifying system can be built in future.

1.1 Problem statement

To review and evaluate different transfer learning models of neural network architecture for classifying and identifying image of various fruits and vegetables.

1.1.1 Hypothesis

"Out of all the different architectures of Convolutional Neural Networks using the transfer learning approach, VGG16 provides the better and efficient results".

1.1.2 Research Questions

- How efficient will be the built models, using transfer learning architecture of CNN when compared to a fully- self trained CNN?
- Which will be the best trained model when compared to convolutional neural network?
- Among the given transfer learning models which all models will prove to be advantageous as per their nature and given specifications?
- Are there any similarities among the trained network models with respect to the fruit or vegetable class predictions?

2 Literature Review

Till date there have been diverse solutions proposed for automation in order to tackle problems in the agricultural sector. In this section we will be discussing the evolution of image classification from basic computer vision to use of machine learning algorithms, deep learning techniques and use of transfer learning approaches for multi-class image classification.

2.1 Use of Machine Learning Techniques (Non-Neural Network) in Image Classification

The history of image classification is huge starting from 1958, with lot of evolution in the techniques and approaches in the field of Computer Vision and AI. One such successful attempt to produce recognition system for trainable product 1D system ("VeggieVision"),is given in article [Bolle et al., 1996], where The application consists of an integrated image and scale device with a convenient to use interface. When a product is put on the scale a image was taken and variety of features, colour, texture (shape, density) were extracted which were further contrasted with "signatures" stored that were collected through prior device training (either online or off-line). Based on the classification accuracy, from a number of options chosen by the machine, the final decision was made either by the machine or by a person with the classification accuracy of 84% for the first option and 95% for the top four recommendations. However, this system was not efficient enough for the real time use as it faced lot many drawbacks as the training and testing datasets were from the same store and when different datasets where used the accuracy declined.

In paper [Rocha et al., 2010], the authors introduced a new approach which was based on feature fusion techniques which used less number of training data with dataset consisting of 15 class type and 2633 images, to option high accuracy. In the places like warehouse or the grocery stores, the proposed fusion method is used for classifying fruits and vegetables in a multi-class image classification. The results show that the solution reduced the classification error by 15% points in relation to the baseline. Automated yield predicition is very much beneficial for the farmers as well as the agricultural units. In order to achieve certain prediction before the crop yield is crucial, based on the previous methods it depends on many feature extraction methods such as shape, colour, hue, size etc. and these features could vary from one fruit to another and in fact one image to another of the same fruit class. With advancement in the Machine Learning techniques these limitations were overcome. However, the evaluation techniques were mostly based on the models accuracy. There are many other evaluation techniques which are as important as accuracy and analysing those factors gives better results for the proposed model. The paper [Yamamoto et al., 2014] uses the recall and precision values along with the accuracy in order to evaluate there proposed model. Author introduced a method to precisely identify tomato fruits in three states (mature, immature and infant) on a plant. By using image segmentation on the shape, size, colour and texture of the image, classification algorithm was modeled. Finally, the evaluation matrix used were precision and recall which was 0.88 and 0.80 respectively.

With the further enhancements in the techniques, different approach were used in image classification one such is seen in paper [Sa et al., 2016], a bounding box annotation is used instead of pixel level annotation and hence, it is one of the easier method to implement. Support Vector Machine (SVM) is best for high dimensional data also in order to classify the fruit into two classes, as SVM works only on dichotomous target variable, tomatoes

are classified into infected and uninfected classes. Principal Component Analysis (PCA) is cost efficient when compared to SVM or other algorithms and same is explained in paper [Semary et al., 2015], mainly used to avoid the dimensional issues.

Further in order to overcome the complications in automatic classification of fruits using computer vision researchers in paper [Zhang and Wu, 2012] suggests an method which is similar to the concept of kernel support vector machine (kSVM) a multi-class classification method. Where in the fruit images were taken and there backgrounds were removed using split-and-merge algorithm. Then the shape feature, color histogram and texture are extracted from each image further reducing its dimensions by using Principal Component Analysis (PCA). Three types of multi-class SVMs along with three types of kernels were developed, (i.e., Winner-Takes-All SVM, Max-Wins-Voting SVM, Directed Acyclic Graph SVM and linear kernel, Homogeneous Polynomial kernel, aussian Radial Base kernel respectively); In order to reduce the feature vectors 5-fold stratified cross validation function was used to train the SVMs, as data and the dataset consisted of 1,653 images corresponding to 18 classes. The findings showed that the Max-Wins-Voting SVM with the Gaussian Radial Basis kernel got the highest 88.2 % classification accuracy and Directed Acyclic Graph SVMs executes fastest. However, in order to improve the accuracy and precision the authors of [Zhang and Wu, 2012] came up with new approach in 2016 as given in paper [Zhang et al., 2014] . The new algorithm had weight optimization feature algorithm based on Fitness-Scaled Chaotic Artificial Bee Colony (FSCABC) and Feed forward Neural Network (FNN). In comparison to the previous approach used in [Zhang and Wu, 2012], FSCABC-FNN gave classification accuracy of 89.1% on the same dataset.

Authors in [Kamilaris and Prenafeta-Boldú, 2018] shows a review on 40 studies using profound learning methods in agriculture and food processing sector. By comparing several techniques and performing comprehensive analysis on various agricultural issues it was concluded that best results and accuracy were seen by using deep learning techniques in comparison to the existing image processing techniques.

2.2 Use of Deep Learning Techniques (Neural Network) in Image Classification

Deep learning with convolution networks is commonly used for image processing tasks, as convNets/conv2D/cv2 can learn transnational symmetric structures, allow object detection anywhere where the image is located, and can retrieve abstract visual concepts by capturing a hierarchy of increasingly difficult structures. As explained in the research paper [Kamilaris and Prenafeta-Boldú, 2018], the starting layers learn basic spatial shapes, like borders, whereas the subsequent layers acquire more abstract object representation, like shape. A unique approach is shown in article [Cheng et al., 2017], where algorithm for image classification of apple fruit under sun light in the orchard is experimented. The uniqueness of this solution is that two back propagation neural network (BPNN) models for early yield estimations was developed with the fruit features paired in four attributes (number of fruit, single fruit size, area of fruit clusters, and foliage leaf area) which can be a very useful enhancement in the field of agriculture.

In 2012, a Large Scale Visual Recognition Challenge 2012 (ILSVRC2012) was conducted as featured in [Russakovsky et al., 2015a], the team which won the challenge by huge difference in accuracy of 85% in comparison with usual Support Vector Machine algorithm with accuracy of 74% made a huge difference in the field of image classification and a hit in the revolution of deep learning. The wining team was AlexNet [Krizhevsky et al., 2012].

There are many such proofs such as explained in paper [He et al., 2016] which justify that the implementation of the Convolutional Neural Networks(CNNs) has significantly improved classification accuracy and results from papers [Tóth et al., 2016, Yu et al., 2017] shows that CNNs were used to identify plants and showed improved accuracy in classification compared to many other methods, such as SVM or RF.

The major application of using image processing and image classification techniques is in the field of agriculture and one such approach for enhancing the robotic harvesting system is shown in paper/article [Puttemans et al., 2016] where the author improves the fruit detection techniques by using supervised machine learning techniques and adding colour information feature specific to the application instead of hand crafted image filters. This technique was tested to detect apple and strawberry fruits thus developing and automated robot harvesting system and in order to achieve this it was recommended to use an object categorization mechanism based on enhanced cascades of low classification. Another approach for enhancing the automated harvesting system was to build a robot which could pick the grapes at night time as shown in [Xiong et al., 2018]. This grape picking robot was programmed in three phases; First, in order to identify the colour in the night time exploratory analysis was done on the R component of RGB image, by rotating the R component. Second, after analysing the R component, using C–V level-set model along with morphological treatment background of the image is removed. Third, in order to pick the fruit precisely above the stem depending on the growing features of the grape stem Hough line detection method was used. This process resulted in 91.67% accuracy [Xiong et al., 2018]. The limitation of this system were it couldn't identify overlapping fruits and moving disturbance occurred.

Handcrafted feature based computer vision techniques are always less efficient than the neural network techniques along with data argumentation features. One such example is explained in paper [Zhang et al., 2019], the authors used a 13-layer convolutional neural network (CNN) with image rotation, Gamma correction and noise injection data argumentation techniques which helped in extending the training data from 1800 to 63,000 and gave an average accuracy of 94.94% which was better than the five state-of-the-art techniques. The limitations of this experiment are, when the classifier was tested on as realistic images from supermarket the overall accuracy was reduced by 5%. The enhancement techniques in deep neural network have made a huge change in object classification and recognition. In order to evaluate these factors the Pascal Visual Object Classes (VOC) challenge on classification, detection, segmentation, action classification, and person layout was conducted from 2008-2012 and article [Everingham et al., 2015] gives the review on these challenges. VOC provides there own image dataset and conduct these competitions. In order to evaluate these algorithms on VOC dataset certain unique techniques were used such as: A boot strapping tool for evaluating whether or not there are substantial variations in the efficiency of two given algorithms; A consistent average precision for measuring results across groups of equal amounts of positive instances; A method of classifying output analysis through several algorithms to define a difficult and simple image scan; And the use of a combined classifier over the provided algorithms to calculate their compatibility and cumulative output. All the errors depending on there conditions were considered. By exploring different methods in order to fruit identification/classification, it can be said that deep neural networks typically offer a better result in prediction, as explained in [Duong et al., 2020] limitations comes at a price such as execution time, training time and requirement of more number of training images and features. In order to achieve this prohibition of cost in training and test, CNNs needs to be scaled up and

modified with which will lead to new architectures.

2.3 Use of Transfer Learning in Image Classification

To minimize the weakness of CNNs the best approach is to use transfer learning techniques. As explained in article [Koirala et al., 2019], it is common practice to employ transfer learning (also known as or fine tuning) on a network pre-trained on a large dataset (e.g., ImageNet) for efficient and more stable training of deep learning models. Transfer learning enables original parameters (convolution weights) to be re-used from a model trained on large datasets to test new models from comparatively smaller training images. The available datasets such as ImageNet provide labeled image data of common objects, free of charge for training and optimizing models for object detection [Koirala et al., 2019].

A faster and more precise method for identification of fruit by using the transfer learning technique is provided in paper [Huang et al., 2019]. With the use of Mobilenet network a method to decrease the parameters and computation price in the training process was built, which was further tuned based on DSConv. The network introduced performs depthwise separable convolution with lighter element to reduce the scale of the vanilla network and increase efficiency by adjusting the regional deptwise convolution. They achieved accuracy of 97%, by dividing the standard convolution into 2 steps (deptwise convolution and pointwise (1*1) convolution, like a factorizing convolution form. Transfer learning methods was also used to further achieve the better accuracy by replacing the average pooling by global deptwise convolution layer helped and in order to avoid over-fitting issues and long training time, as all these modifications are only possible by using transfer learning. Another example where MobileNetV2 with transfer learning technique is used for image classification is explained by author in [Xiang et al., 2019], the researchers conducted experiments on MobileNetV1, InceptionV3 and DenseNet121 networks with fruit image dataset consisting of 3670 images of 5 class categories. The best accuracy of 85.12% was obtained by using MobileNetV2 network, wherein the top layer of the network was replaced with conv2d layer along with dropout in order to reduce over-fitting, extraction of features was done by pre-trained ImageNet dataset as a base network and to classify trained features a Softmax classifier was used. However, the proposed method was only suitable for limited devices with smaller class and mobile devices. However, in paper [Femling et al., 2018] MobileNet architecture is used in automated recognition of fruits and vegetables by self-service systems for super markets. It was concluded that MobileNet gave fast classification with accurate prediction results with accuracy of 97% when compared to InceptionV3 architecture. The drawback of proposed model was the propagation time was larger than the network size.

Like Mobilenet there are many other architectural methods built using transfer learning approach in the agricultural sector in order to help the farmers to improve there productivity by identifying the health condition or quality of fruits/vegetables. One such method is used in paper [Nikhitha et al., 2019], authors introduced an application to classify the level of the disease and grade them accordingly. InceptionV3 model was used and retrained using Imagenet dataset on total of 101 fruit classes with training set of 522262 images and testing set of 17540 images. The model was modified on the outer two layers which is the bottleneck layer so that the cache values of individual training image can be stored and process speed can be increased at the time of validation. The drawback of process in [Nikhitha et al., 2019] is no valid explanation was given on the factors used to differentiate the images based on disease as the output of proposed method was given in 2 steps; first is

Classification of fruits and second is Percent of disease identification. Another approach in the field of image classification for classifying fruits is Retraining_Inceptionv3 model, which is a transfer learning method of Inceptionv3 which uses google training dataset "Imagenet" is introduced in [Jeong and Yoe, 2018]. In order to execute the experiment image data of 3 classes (Red apple, wounded apple and Green apple) were taken as 30 images per class, the resulting accuracy of 73% was achieved. However, the repeated evaluations proved that the classifier accuracy can be increased if the number of training steps and training size is increased provided it depends on the computation speed of the individual computer/system. By the use of transfer learning and fine tuning techniques image classification accuracy can be significantly increased, which is well shown in the above discussions and paper [Siddiqi, 2019] supports our statement. Researchers used Inceptionv3 and VGG16 transfer learning models in order to conduct the experiment. The dataset used for the purpose is [Mureşan and Oltean, 2018] the Fruits 360 dataset with 48,249 fruit images of 72 classes. Also, to prove there hypothesis a convolutional neural network model of 14 layers was designed and tested on the same dataset which resulted in 96.79% of accuracy. Whereas, by using pretrained models like VGG16 and InceptionV3 gave accuracy of 99.27% and 98.10% of accuracy [Siddiqi, 2019]. Hence, it was concluded that these were the best results received when compared to approaches in previous methods discussed in research articles such as, [Kamilaris and Prenafeta-Boldú, 2018, Zhang et al., 2014, Nikhitha et al., 2019, Jeong and Yoe, 2018, Mureşan and Oltean, 2018]. The Challenge or drawback here in [Siddiqi, 2019] was the dataset used had clear images, which is not applicable in real time and if the algorithm is used for testing imperfect image the desired results will not be obtained. Another experiment which shows VGG16 gives the best results is shown in article [Hossain et al., 2018], researchers here wanted to built an automated fruit recognition system. For the purpose 2 models were built (CNN with data argumentation and VGG16) and tested on 2 different datasets, one with clear fruit images (15 classes of fruits) and other consisting of blur fruit images which were difficult to classify (10 classes of fruits). Accuracy of 99.49% was achieved by both the models on the first dataset with clear images. Whereas, VGG16 gave 96.75% and CNN gave 85.43% of accuracy when tested on the second dataset with unclear images.

Deep learning and transfer learning has always been the best combination in the field of fruit processing industry, author in [Adigun et al.,] shows how an automated system for grading of apples is created using CNN and ResNet50 (transfer learining process for convolutional base). The introduced system in the paper [Adigun et al.,] had 2 models an Apple Checker Model (ACM-used to identify apple or not) built using ResNet50 and Apple Grader Model (AGM-used to grade the apples in 4 categories) built using CNN. ACM resulted in 100% accuracy where as, AGM gave 99.89% accuracy. However, the classification model was based on binary classification.

For any automated system it is necessary that the proposed model should be efficient and effective, the above discussed methods are either efficient or effective, which makes a drawback in the real time scenario. However, researchers in paper [Hossain et al., 2018] introduces a classifier model which is both effective and efficient i.e., EfficientNet and MixNet, the major advantage on these networks are they can be deployed on smaller devices with minimal computing tools. Experiment was conducted on Fruit 360 dataset [Mureşan and Oltean, 2018] with 48,905 training images and 16,421 testing images of 95 fruit classes. They concluded that EfficientNet-B5 and MixNet-Small gave the best results using the transfer learning logic.

Transfer learning techniques are not only being applied using ImageNet weights, there

are other architectures as well which are being used one such architecture is called as Orchards. To enhance agricultural activities such as yield mapping and robotic harvesting, an reliable model was built using transfer learning on Orchards architecture in [Bargoti and Underwood, 2017a]. Experiments were conducted on apple, mango and almond fruit type of images present in orchards by using state-of-the-art detection design called as Faster-RCNN. Experiments were also conducted to compare the results of transfer learning using orchards and ImageNet, unfortunately using both the weight transfer methods did not show any major difference in the performance or accuracy of the model. However, better results were achieved by using flip and scale argumentation features of data argumentation technique. The overall analysis resulted in the improvement of fruit detection for apple and mango with F1-Score of greater than 0.9 [Bargoti and Underwood, 2017a].

One approach towards making the image classification models qualitative is by using novel technique of RGB and near Infrared (NIR) multi-modal fusion information the researchers of [Sa et al., 2016] developed an high-performance real time fruit recognition system based on Deep Convolutional Neural Networks (DCNN) system that can be quickly trained with a less number of images which is pre-trained on ImageNet dataset [Russakovsky et al., 2015b]. In order to evaluate the results, F1 scores of implemented RGB and NIR multi-modal networks were compared with fine tuned VGG16 network. It was shown that by using the novel approach better results are obtained over a single DCNN [Sa et al., 2016]. Most of the proposed models have certain limitations due to training and testing done on limited number of images/dataset without considering the external features, one approach used in [Hussain et al., 2018] were the researchers have tried to overcome these limitations by creating a new fruit dataset with 15 different classes consisting of 44406 images, keeping the real-world conditions into consideration. Each input image was converted into grey scale and given to the Deep Convolutional Neural Network of 5 layers. The overall accuracy achieved was 99%. As the technology enhances one novel approach was explained in paper [Khan and Debnath, 2019] which was used for both fruit recognition and detection called as FDR model. This model had reduced the training time and difficulties during training. The structure of this CNN based recognition FDR model, was a combination of variety of convolution, sub-sampling and fully connected layers in a reasonable manner which made the recognition model more robust and trustworthy. This method was different from conventional window-based detection based function or sliding which distinguished each object very well from an image. In addition, the FDR model delivers improved results in fruit region identification and has archived an average precision rate of 0.9875 using dataset [Mureşan and Oltean, 2018] images.

2.4 Outcomes of Literature Review

As we discussed several novel techniques and experiments conducted by researchers previously in the field of multi-class image classification, in this paper experiment are conducted on the Fruit 360 [Mureşan and Oltean, 2018] which is the largest dataset available. Models with transfer learning techniques will be implemented which are inspired by the above discussed research papers such as, [Huang et al., 2019, Siddiqi, 2019, Adigun et al.,] along with CNN architecture as shown in paper [Zhang et al., 2019] and using pre-processing techniques as used in paper [Hussain et al., 2018]. Each model will be evaluated and results will be compared and discussed in the later part of this article.

3 Neural Network Theory and Architecture

This section gives the brief introduction and structure of the Convolutional Neural Networks and its architecture used in this study.

3.1 Convolutional Neural Network (CNN/ConvNet)

Convolutional Neural Network is a type of Deep Learning algorithm which are made up of neurons and takes the input as image, assigns weights and biases which are trainable. The drawback of regular neural network is overfitting as the input image is not completely scaled up. CNNs are most popular in image classification and object detection applications. The neurons of CNN are arranged in 3D (three dimension) format, i.e., width, height and depth, as shown in figure below.

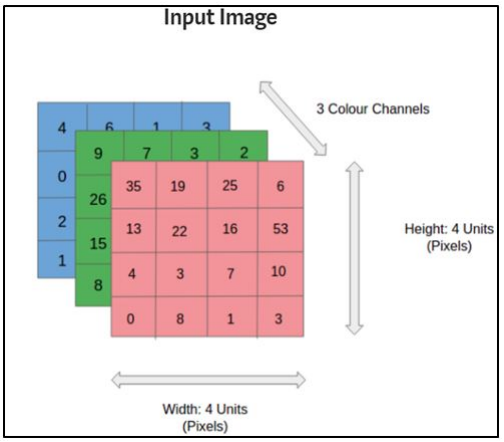

Figure 1: RGB Image [Saha, 2018]

Every image when fed to a machine is converted into a matrix form and further it gets flattened into single vector format arranged in depth dimension. Therefore whenever a complex image or thousands of images are given to the machine at the same time, the computation time and complexity increases there by CCN/ConvNet is beneficial here, which vectorise the images in simpler format by keeping up its feature values.

3.1.1 ConvNet Layers

Convolutional architecture is built using series of layers. The three important layers are convolutional layer, pooling layer and FC (fully-connected) layer. These layers converts one block of activation to another by means of non-linear function. Figure 2, shows a simple CNN layer sequence for classifying a car image.

Let us consider this as an example with image size 32x32x3 for our understanding of the layers.

- **Input Layer-** Here the input image is fed and its pixel values are stored in the form of width size of 32, height size of 32 and 3D colour which is B, G, R (Blue, Green Red) format.

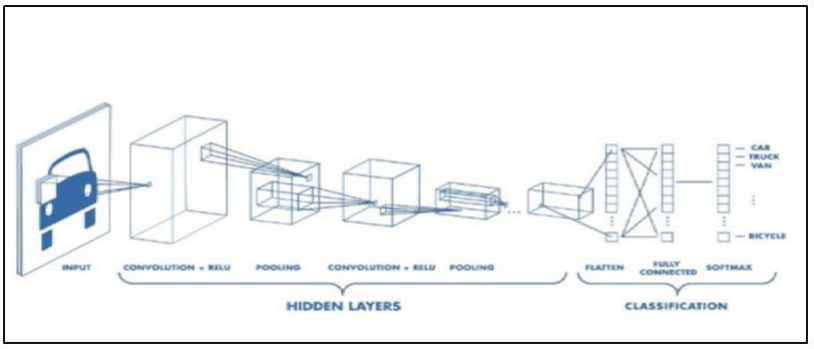

Figure 2: CNN Layer sequence

- **Hidden Layer-** This layer consist of convolutional layer with activation functions and pooling layer.
  - **Convolution Layer:** The basic functionality of using Conv layer is to capture low-level features and as we increase the number of Conv layers it can capture the high-level features as well, these features includes edges, color, gradient orientation and we obtain these features by performing dot product between there weights.
  - **Relu Layer:** In order to keep the volume size same relu layer is used which generates a activation function. It is called as Rectified Linear Units. When it gets any negative feedback, the function returns 0 and it returns the value for every positive value x. So you can write it as f(x)=max(0,x).
  - **Pooling Layer:** This layer is responsible for down-sampling or decreasing the dimensions of the image features weight and height which helps in reducing the computing power needed for the data processing by reducing the dimensions. In addition, it is useful for selecting significant features which are symmetric, rotational and positional while preserving the model's efficient training cycle.
- **Classification Layer-** Classification is the final layer in the architecture, which consists of Flatten layer, FC layer and Softmax layer.
  - **Flatten Layer:** This layer acts as a bridge between convolution layer and FC layer, as it is responsible for converting 2D matrix values into a single column vector format.
  - **Fully Connected (FC) Layer:** This is the important layer which is responsible to compute the volume size with respect to the class score. Inputs from the vectorized features are taken and as per the weights assigned corresponding class or label category is predicted.
  - **Softmax Layer:** When dealing with multi-class classification, softmax layer is used as the final output layer. The word derives from the softmax activation function, which takes a number of predicted classification scores as its input and distributes them into levels in the range of 0 and 1 of which the sum is 1, which in turn gives the output class with the high probability value.

3.1.2 Different ConvNet Layer Configuration

There are different CNN architectures available that were key in creating algorithms that power and control AI as a whole, over the coming years. Some are termed as classic network architecture, such as: LeNet-5, AlexNet, ZFNet, GoogleNet, VGG16. And, some are termed as modern network architecture, such as: Inception, ResNet, ResNetXt, DenseNet, MobileNet, NASNet.

- **LeNet-5-** In the 1990's Yann LeCun developed the few promising applications of Convolutional Networks. Among such, the renowned architecture is LeNet, purpose of which was to read zip codes, digits, etc.
- **AlexNet-** AlexNet was devloped by Alex Krizhevsky, Ilya Sutskever and Geoff Hinton and it was the first Convolutional Network whcih was very famous among the other computer vision techniques. In 2012 the AlexNet was reviewed to the ImageNet ILSVRC challenge and received the second runner-up prize. This network had an architecture identical to LeNet, but it was wider, broader and featured Convolutional Layers layered on top of one another.
- **ZFNet-** In 2013, the winner of the ILSVRC was a Matthew Zeiler and Rob Fergus Convolutional Network. Thus it is known as ZFNet short form of Zeiler and Fergus. It was an enhancement on AlexNet by modifying the hyperparameters of the design, in specific by increasing the size of middle convolution layers and keeping the phase and filter size smaller on the first layer.
- **GoogleNet-** In 2014, the ILSVRC challange was won by Szegedy et al. from Google. The key contribution was to create an Inception Module that significantly decreased the number of network variables. GoogleNet uses Average Pooling at the top of the ConvNet rather than Fully Connected layers, removing a massive proportion of variables that don't seem to make a difference. Inception series is part of GoogLeNet and the latest version is Inception-v4.
- **VGGNet-** The runner-up of ILSVRC challenge in 2014 was VGGNet proposed by Karen Simonyan's and Andrew Zisserman. The key achievement of this network was to illustrate that network depth is a vital component of successful efficiency. Their pretrained edition is available for use in plug and play. One disadvantage of the VGGNet is that testing is more costly and requires much more storage and variables (140 M). Most of these variables are in the first fully connected layer, and it has since been found that these FC layers can be extracted without any degradation in performance, greatly reducing the amount of parameters required.
- **ResNet-** The winner of ILSVRC 2015 challenge was Residual Network introduced by Kaiming He et al. It specifically described skip connections, as well as heavy batch normalization usage. At the network end, they have removed the fully connected layer. Presently ResNets are state of the art Convolutionary Neural Network models.
- **DenseNet-** The winner of ILSVRC 2017 challenge was DenseNet. It was jointly devloped by Cornwell University, Tsinghua University and Facebook AI Research (FAIR). Here, every layer receives a "collective knowledge" from all the previous layers. There by, making the network smaller and compact which leads to increase in computational and storage efficiency.

Let us now discuss important architectures which are used in the experimental analysis.

3.2 VGG16

One of the simplest and successful CNN architecture introduced is VGG16 called as "Visual Geometry Group". The uniqueness of this architecture was the depth of the architecture was increased to 16 and 19 layers with very small (3×3) convolution filters. The architecture of the VGG16 consists of 12 convolutional layers, preceding by max pooling layers, then 4 FC layers and ultimately a softmax classifier with 1000-way, which means we can use VGG16 to classify 1000 image classes. Figure 3, shows the basic architecture of VGG16 and figure 4 shows the architectural summary of VGG16.

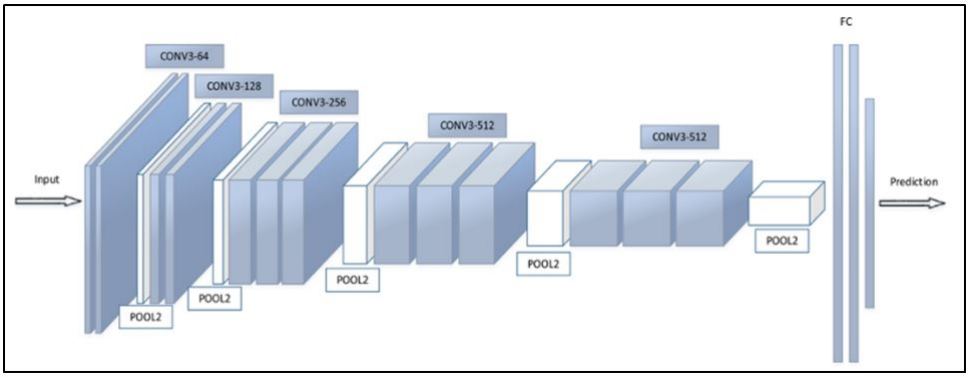

Figure 3: VGG16 Architecture

Layer		Feature Map	Size	Kernel Size	Stride	Activation
Input	Image	1	224 x 224 x 3	-	-	-
1	2 X Convolution	64	224 x 224 x 64	3x3	1	relu
	Max Pooling	64	112 x 112 x 64	3x3	2	relu
3	2 X Convolution	128	112 x 112 x 128	3x3	1	relu
	Max Pooling	128	56 x 56 x 128	3x3	2	relu
5	2 X Convolution	256	56 x 56 x 256	3x3	1	relu
	Max Pooling	256	28 x 28 x 256	3x3	2	relu
7	3 X Convolution	512	28 x 28 x 512	3x3	1	relu
	Max Pooling	512	14 x 14 x 512	3x3	2	relu
10	3 X Convolution	512	14 x 14 x 512	3x3	1	relu
	Max Pooling	512	7 x 7 x 512	3x3	2	relu
13	FC	-	25088	-	-	relu
14	FC	-	4096	-	-	relu
15	FC	-	4096	-	-	relu
Output	FC	-	1000	-	-	Softmax

Figure 4: VGG16 Architecture-Summary Table [Rizwan, 2018]

The highlights of VGG16 architecture are:

- The weights and biases of this network are learnt in total 16 layers.

- 13 Convolutional layers are placed one after another, which converts 224x224x3 size of image to 14x14x512 with kernal size of 3x3.
- The filter size in the convolution layers is following an increasing trend (equivalent to autoencoder-decoder architecture).
- The insightful features are computed through the implementation of max pooling layers at various architectural levels.
- Finally 3 dense layers are stacked one after the other and last dense layer is used for classification output.
- Due to this configuration, the model becomes very large in size, thereby increasing the training time and storage space which becomes drawback of VGG16 architecture.

3.3 Inception

Inception network also called as GoogleNet was created in order to overcome the limitations of other networks and increase its performance, both in terms of speed and accuracy. There are 5 most popular versions of Inception network, which are InceptionV1, InceptionV2, InceptionV3, InceptionV4 and Inception-ResNet. The versions are the enhanced features of the network. Figure 5, shows the Inception module with reduced dimensions.

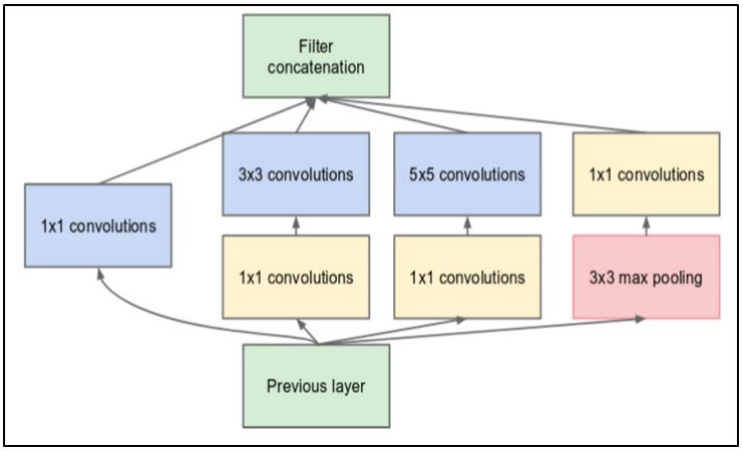

Figure 5: Inception Module [Szegedy et al., 2015]

A series of convolutions are performed at basic units called as "Inception Cell" so that the results can be added and averaged. To save computation time, it uses 1x1 convolutions to lower the depth of the input loop. A series of 1x1, 3x3, and 5x5 filters for each cell is used to learn how to extract features from the input at various scales. Using this dimension reduction, a neural network architecture was built known as InceptionV1 as shown in figure 6.

The highlights of InceptionV3 architecture are:

- InceptionV3 is 42 layers deep network and the computation cost is only about 2.5 higher than GoogleNet (InceptionV1).

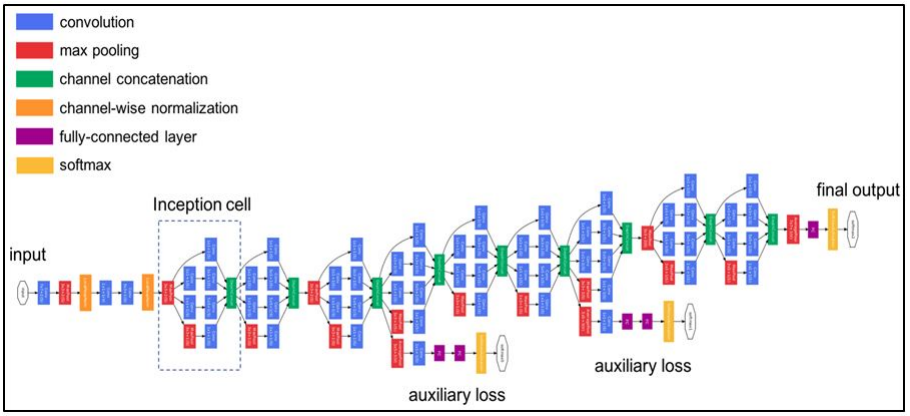

Figure 6: Inception Network Architecture [Szegedy et al., 2015]

- InceptionV3 has two 3×3 convolutions instead of one 5×5 convolution for factorization into smaller convolutions. The objective of the Convolutions factorization is to decrease the number of connections / parameters without reducing system performance. By doing so parameters are reduced by 28%.
- InceptionV3 uses factorization into asymmetric convolutions technique where 3x3 convolution is replaced by one 3x1 along with one 1x3 convolution. So that, number of parameters are reduced by 33%.
- Rather than using 2 auxiliary classifiers, just 1 auxiliary classifier is used above the final 17×17 layer. The reason for using auxiliary classifier in InceptionV3 is it acts as regularizer. Whereas, in InceptionV1 it is used to build a deeper network.
- Usually, downsizing of features is achieved by max pooling layer as seen in other architectures such as AlexNet, VGGNet. But this addition makes the network expensive. Therefore, InceptionV3 uses efficient grid size reduction technique.
- Two sets of feature maps are used, one with Conv stride 2 giving 320 feature maps. And other uses max pooling to get another set of 320 feature maps. Finally these 2 sets are concatenated to achieve an efficient grid size reduction.

3.4 ResNet

Many of the earlier models used deep neural networks in which several convolution layers were layered one after the other. It was assumed that this approach is best for Deep Neural Networks. However, it turned out to be wrong when ResNet was introduced. While understanding on certain problems associated with the other networks such as: (based on[He et al., 2016])

1. Optimization of the network becomes difficult.
2. Deep networks are hard to train because of the notorious vanishing gradient problem / exploding gradients
3. Accuracy of typical networks first exhausts and then degrades.

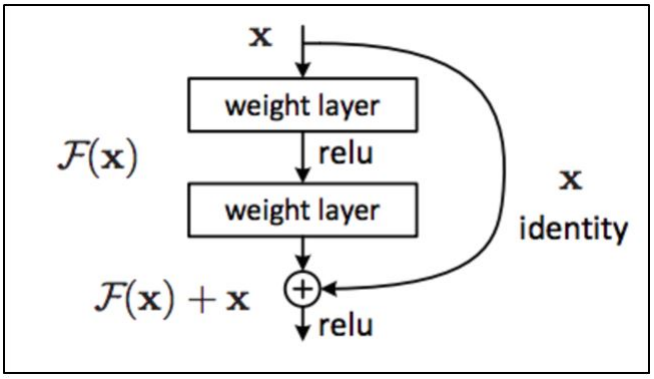

Figure 7: ResNet: Building Block [He et al., 2016]

The figure 7, shows the ResNet building block based on the hypothesis stated in article [He et al., 2016]: Instead of optimizing the, unreferenced mapping H(x), optimizing the residual mapping function F(x) is relatively easy. Which is achieved by using skip connection as shown in figure 7, by copying the activation from shallow layer and setting additional layer to identity mapping. In order to allow such connections or the extension process, it is important to maintain the same dimension of convolutions across the network, hence resnet uses 3x3 convolutional dimension throughout the network.

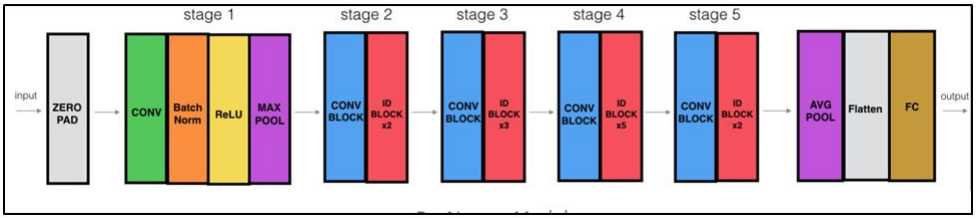

Figure 8: ResNet-50 Architectural Block Diagram

Figure 8, shows the ResNet-50 architectural diagram. The ResNet-50 has almost 23 million trainable parameters. The model consists of 5 stages, each stage is a combination of convolution block and Identity block. All the convolution block and identity block has 3 convolution layers each. Identity function is the easy factor for residual blocks to learn. Adding these residual blocks, did not increase the model complexity. Because, it is just replicating the previous activation to the next layers. Thereby, adding the residual blocks or the skip connections will not hamper the network performance. Instead, the training and learning opportunities to the new or next layers will be increased.

3.5 DenseNet

As the network grows deeper in size it creates a problem for a Convolutional Neural network. This is caused due to increase in distance path of input data transmission and data getting lost or tampered during this transmission from input layer to the output layer. In order to resolve this connectivity issue DenseNet was introduced by researchers as stated in [Huang et al., 2017]. DenseNet uses feature reuse technique and increases the potential of the network, instead of taking representational power from deep or wide architectures. As seen in figure 9, within a dense block, each layer's feature map is joined

to the input of very next layer. Due to this, the immediate layers within the network grasp the features from previous layers. Author states that, "concatenating feature-maps learned by different layers increases variation in the input of subsequent layers and improves efficiency" [Huang et al., 2017].

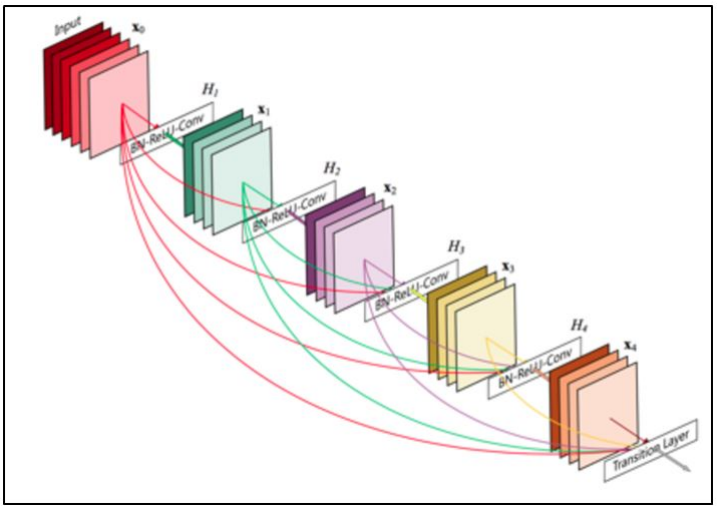

Figure 9: DenseNet with 5 Layers [Huang et al., 2017]

DenseNet Comprises of two important blocks; Dense Block and Transitional Layers. Dense Blocks are the ones where the dimensions of the feature maps remains constant within a block, but the number of filters keep changing between them. Transitional layers are the layers between the blocks which controls the downsampling by using a batch normalization technique with 1x1 convolution layer and a 2x2 pooling layer.

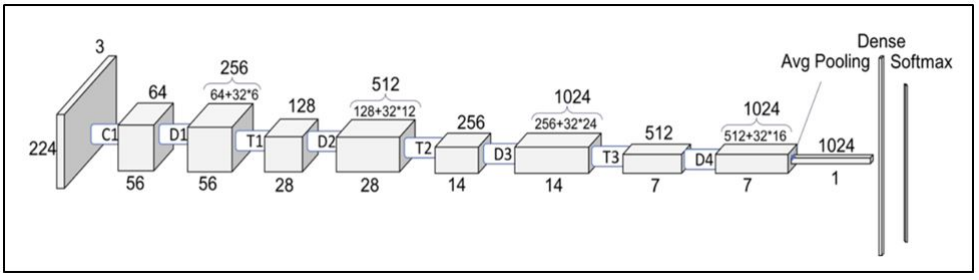

Figure 10: Dense-121 Architectural Block Diagram

Figure 10, shows the basic architectural block diagram of DenseNet-121. Where the Dx blocks are the Dense Blocks and Tx blocks are the transitional layers/blocks. The measures under each block represent the size that is width and depth of the layers. Whereas, the top numbers shows the feature maps dimension. There are two important outcome of this architectural arrangement;

1. The volume within a Dense Block remains constant.
2. The volume and the feature maps are divided by 2 after every Transitional Block.

This is achieved as seen as per the calculation in figure 10, each layer gets added with its previous layer and 32 new features are mapped. Thereby, after 6 layers we get 256 from 64. As the Transition block executes 1x1 convolution with 128 filters, along with 2x2 pooling with 2 strides. Hence, the result is volume size is divided and feature maps by 2. As per the research by the author, it is said that DenseNet gives better results when compared to ResNet, even though both the architectures are similar, in fact the DenseNet is just a copy of ResNet just that the dense block is swapped as the repeated unit.

3.6 MobileNet

MobileNet is also called as Depthwise Seperable Convolution network. This network was introduced specifically for the mobile based and embedded vision applications, due to its compact model size and complexity. Figure 11, shows the basic building block of a Depthwise Seperable Convolutional Network, which consists of 2 main blocks placed next to each other they are; Depthwise Convolution block and Pointwise Convolution block. By using this combination the network can perform a single convolution on individual colour channel (RGB) instead of merging all the three (R, G and B) together and flattening it.

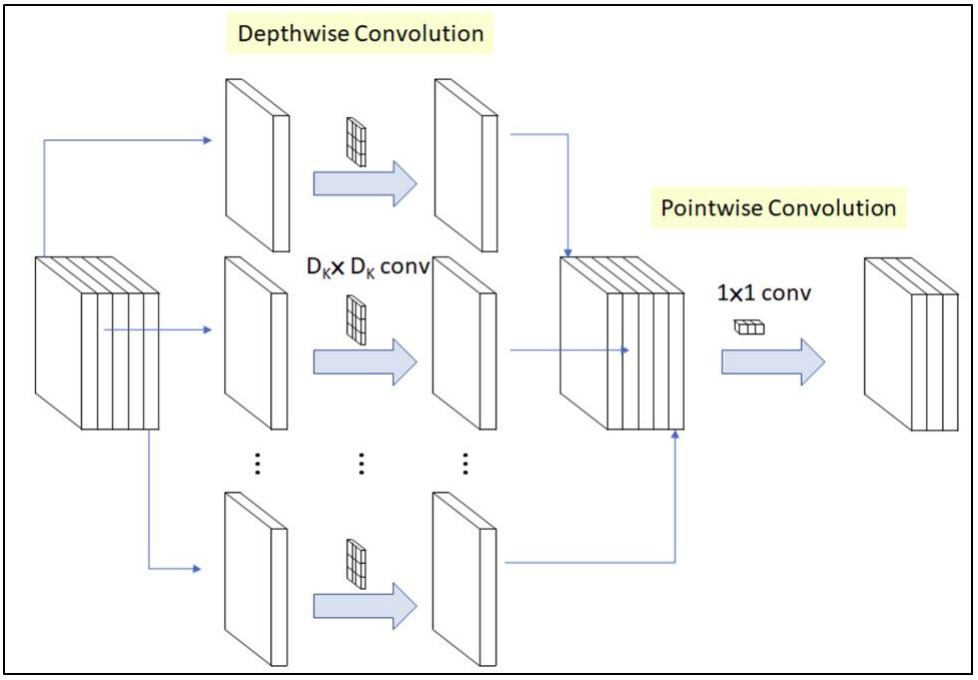

Figure 11: MobileNet- Depthwise Seperable Convolutional Block Diagram

1. Depthwise convolution is the channel-wise Dk×Dk spatial convolution, which applies one filter to every input channel individually.
2. Pointwise convolution is used to change the channel dimensions. This is achieved by applying a 1×1 convolution, so that the outputs from deptwise convolution can be combined.

In other convolution networks, the filtering of input process and mixing of the outputs are performed in one step process. Whereas, in MobileNet this process is divided into two parts called as factorizing where we have individual layers for filtering and mixing/combining

respectively. This is important feature which makes the model size smaller and compact [Howard et al., 2017].

Type / Stride	Filter Shape	Input Size
Conv / s2	$3 \times 3 \times 3 \times 32$	$224 \times 224 \times 3$
Conv dw / s1	$3 \times 3 \times 32$ dw	$112 \times 112 \times 32$
Conv / s1	$1 \times 1 \times 32 \times 64$	$112 \times 112 \times 32$
Conv dw / s2	$3 \times 3 \times 64$ dw	$112 \times 112 \times 64$
Conv / s1	$1 \times 1 \times 64 \times 128$	$56 \times 56 \times 64$
Conv dw / s1	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 128$	$56 \times 56 \times 128$
Conv dw / s2	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 256$	$28 \times 28 \times 128$
Conv dw / s1	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 256$	$28 \times 28 \times 256$
Conv dw / s2	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 512$	$14 \times 14 \times 256$
$5\times$ Conv dw / s1	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
$5\times$ Conv / s1	$1 \times 1 \times 512 \times 512$	$14 \times 14 \times 512$
Conv dw / s2	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
Conv / s1	$1 \times 1 \times 512 \times 1024$	$7 \times 7 \times 512$
Conv dw / s2	$3 \times 3 \times 1024$ dw	$7 \times 7 \times 1024$
Conv / s1	$1 \times 1 \times 1024 \times 1024$	$7 \times 7 \times 1024$
Avg Pool / s1	Pool 7×7	$7 \times 7 \times 1024$
FC / s1	1024×1000	$1 \times 1 \times 1024$
Softmax / s1	Classifier	$1 \times 1 \times 1000$

Figure 12: MobileNet Body Architecture [Howard et al., 2017]

Figure 12, shows the detailed layers of the Mobilenet architecture. As seen it consist of 30 layers in total. The Pattern which is followed is as below and same is seen on the figure 12.

- Starting with a convolution layer with stride 2.
- Followed by a deptwise layer with stride 1.
- A pointwise layer in order to double the input channel.
- Then deptwise layer with stride 2.
- Again followed by a pointwise layer to double the channels.
- Above steps are repeated as seen. Then the final layers are;
- A average pooling layer to calculate the average value of the feature maps.
- Fully Connected layer with a Softmax classifier to classify up to 1000 image class.

Many models provide same accuracy as MobileNet provides, but the advantage is its smaller and compact network size. Due to this it is low in maintenance and performance speed is high. For any network the speed and power is calculated by analysing proportionality between the MACs (Multiply-Accumulates) factor, which is the measure of used multiplication and addition operations within the network.

4 Methodology

4.1 Introduction

This section will explain the methodology followed in this study. The ultimate aim of this study is to built an efficient and effective automated multi-class fruit and vegetable classifier system, which can be used in the food processing industries and supermarkets and making humans life easy. However, currently we are more focused on identifying, investigating and evaluating the Convolutional Neural Network models and their architectures in the form of Transfer Learning. By conducting series of experiments, so that the best model can be achieved and used for our ultimate application goal.

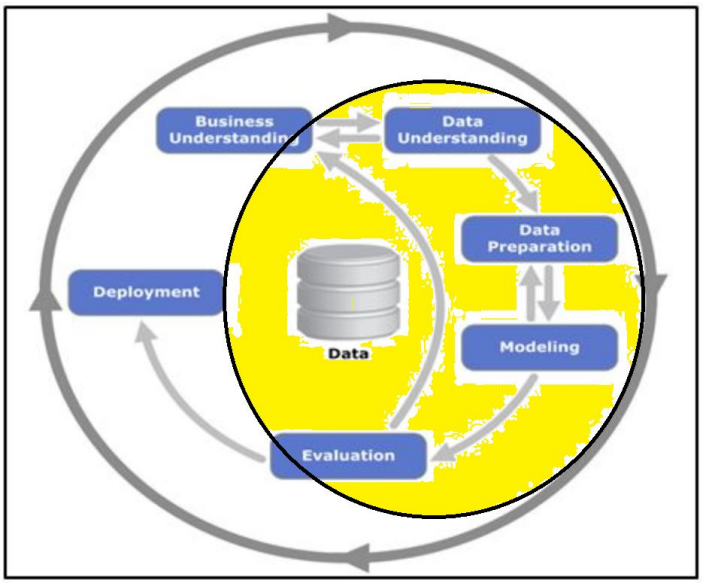

Figure 13: Phases of CRISP-DM Reference Model (modified) [Shearer, 2000]

As seen in figure 14, concept of CRISP-DM reference model is used as a base methodology for this study. Wherein, the highlighted sections is the one which is focused more in this paper. Apart from Business Understanding and Deployment, other phases such as Data Understanding, Data Preparation, Modeling and Evaluation are the phases on which this study revolves around. Below are the steps which have been carried on each and every phase of this methodology.

- **Business Understanding-** Before performing these experiments, the main purpose and the need for this study was understood and kept into consideration. As the purpose is to built an automated application in order to classify the 1000's of variety of fruits and vegetables, so that manual efforts and human errors can be avoided.
- **Data Understanding-** In order to perform classification task on images, understanding of data is very much important, i.e., the type of fruit and vegetable images to be selected and captured, by keeping all the details into consideration such as the angle, location, background, focus, camera lens quality etc.
- **Data Preparation-** When dealing with image as input and applying image processing techniques, this phase in the entire study act as a root base because the kind

of techniques being used in order to pre-process the image pixels and convert them into the matrix format so that the machine can understand. In order to achieve this data argumentation techniques are used.

- **Data Modeling-** The experiments conducted in the study is mainly focused on the data modeling phase. As, the aim is to review and find out the best Neural Network model. The experiments are conducted on different models such as; CNN, VGG16, ResNet50, MobileNet, InceptionV3, DenseNet121, Xception and Inception-ResNetV2.
- **Evaluation-** When there are series of experiments being conducted on different models, evaluation techniques used to decide the best suitable model for the application is very important. Hence, in this study the models are being evaluated on the factors such as; Accuracy, Precision, Recall, F1 Score and also the training time of the model.
- **Deployment-** Deployment phase of the methodology is not being implemented in this particular study. However, when the application is built as part of future work, deployment steps will be performed.

4.2 Proposed Module Design

This section shows the proposed module design for the study and gives the brief overview on the procedure to execute the plan, as seen in the figure 15.

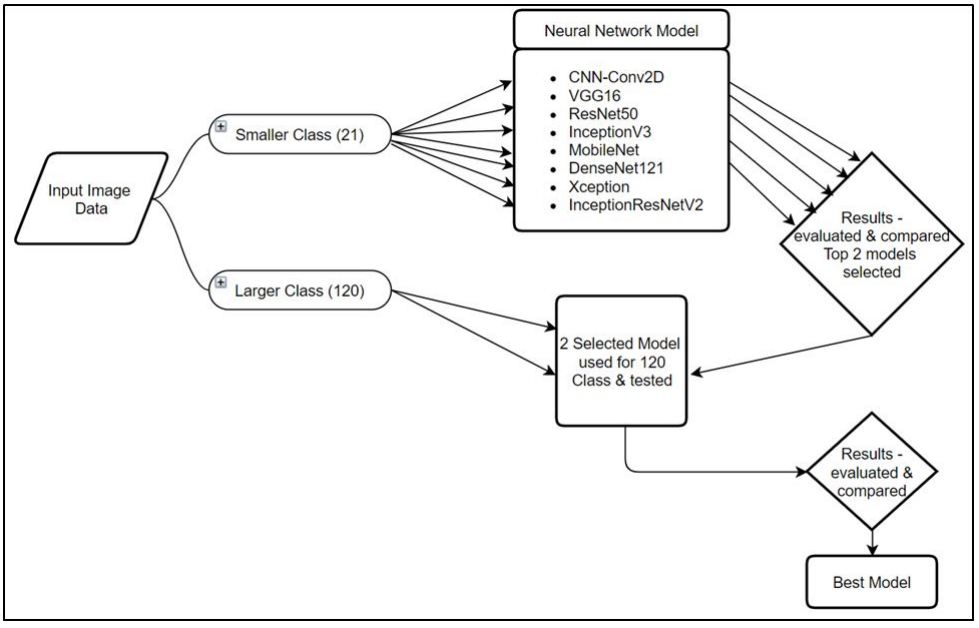

Figure 14: Module Design for the Study

The flow of this study is as follows:

1. Dataset is extracted from the open source website [Oltean, 2017], which is stored in the local directory.

2. As the input image classes are huge, experimenting on all the class together on every model was difficult considering the system configuration. Hence, it was divided into two groups, one with 21 image classes and other with whole 120 classes.

3. The smaller group class was given as input to the neural network, in order to test the best classification model. The selection of model were considered from the official Keras application library [Keras, 2020]. The following models were individually trained for classification:

    - CNN- Convolutional Neural Network
    - VGG16
    - ResNet50
    - MobileNet
    - InceptionV3
    - DenseNet121
    - Xception
    - InceptionResNetV2

4. Upon successful completion of training and testing of the models individually. There respective classification scores such as Accuracy, Precision, Recall, F1 Score and Training Time were evaluated and compared. There by, the best 2 models were selected.

5. In order to further complete the experiment, the top 2 models were individually trained again but using the whole dataset of 120 class images.

6. The results of these models were again evaluated and compared based on same factors as before. Thereby, giving the best model for multi-class image classification.

4.3 Dataset Used

The dataset which is used for this experiment is "Fruit 360" which was introduced as open source dataset by researchers in [Mureşan and Oltean, 2018]. They created the dataset by recording the video of an individual fruit for 20 seconds, which was placed on a shaft of a motor with lower speed of 3RPM with a white paper used as background. Thereby, capturing the image frames out of it. Further, due to external imbalance in the background conditions, background had to be removed from the images using flood fill type of algorithm. Where each pixel is marked starting from the edge until all the pixels are covered, finally the marked ones are considered as background and left over ones were considered as the fruit object. Images were scaled as 100x100 pixel size, in order to avoid any ambiguity between two fruits that may have same colour and shape but different size, such as, apple and a cherry. The initial phase of this dataset contained 38409 images with 60 different fruit types. It was made available on open source platforms such as kaggle and Github.

For performing the experiments, dataset is picked from kaggle website [Oltean, 2017]. The total number of training images are 60498 and testing images are 20622 from 120 classes, which means 120 types of fruits and each fruit type has approximately 460 training images and 160 testing images.

The 120 types or classes of fruits are as below and same is shown in figure 13:

Apple Braeburn, Apple Crimson Snow, Apple Golden 1, Apple Golden 2, Apple Golden 3, Apple Granny Smith, Apple Pink Lady, Apple Red 1, Apple Red 2, Apple Red 3, Apple Red Delicious, Apple Red Yellow 1, Apple Red Yellow 2, Apricot, Avocado, Avocado ripe, Banana, Banana Lady Finger, Banana Red, Beetroot, Blueberry, Cactus fruit, Cantaloupe 1, Cantaloupe 2, Carambula, Cauliflower, Cherry 1, Cherry 2, Cherry Rainier, Cherry Wax Black, Cherry Wax Red, Cherry Wax Yellow, Chestnut, Clementine, Cocos, Dates, Eggplant, Ginger Root, Granadilla, Grape Blue, Grape Pink, Grape White, Grape White 2, Grape White 3, Grape White 4, Grapefruit Pink, Grapefruit White, Guava, Hazelnut, Huckleberry, Kaki, Kiwi, Kohlrabi, Kumquats, Lemon, Lemon Meyer, Limes, Lychee, Mandarine, Mango, Mango Red, Mangostan, Maracuja, Melon Piel de Sapo, Mulberry, Nectarine, Nectarine Flat, Nut Forest, Nut Pecan, Onion Red, Onion Red Peeled, Onion White, Orange, Papaya, Passion Fruit, Peach, Peach 2, Peach Flat, Pear, Pear Abate, Pear Forelle, Pear Kaiser, Pear Monster, Pear Red, Pear Williams, Pepino, Pepper Green, Pepper Red, Pepper Yellow, Physalis, Physalis with Husk, Pineapple, Pineapple Mini, Pitahaya Red, Plum, Plum 2, Plum 3, Pomegranate, Pomelo Sweetie, Potato Red, Potato Red Washed, Potato Sweet, Potato White, Quince, Rambutan, Raspberry, Redcurrant, Salak, Strawberry, Strawberry Wedge, Tamarillo, Tangelo, Tomato 1, Tomato 2, Tomato 3, Tomato 4, Tomato Cherry Red, Tomato Maroon, Tomato Yellow, Walnut.

Figure 15: Fruit Images of 120 Types

4.4 Transfer Learning System Architecture

In this section we will be looking into the complete transfer learning system architecture being used in the experiments conducted. Before that, let us understand what exactly is transfer learning and benefits of using it.

Transfer learning is a process where we can resuse the existing knowledge. This is a method where a state-of-art model is used, which is trained on a huge dataset, with desired pre-processing techniques for longer training duration. It can simply be implemented by first train a base network on a base dataset and task, and then we re-purpose the learned features, or transfer them, to a second target network to be trained on a target dataset and task. This process is useful if the target dataset features used is general, which means

suitable to both base and target tasks, instead of specific to the base task.

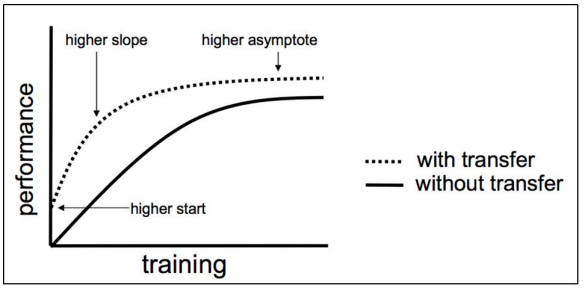

Figure 16: Performance Graph- with without TL [Torrey and Shavlik, 2010]

There are 3 important advantages of using transfer learning, figure 16 shows the training v/s performance graph with and without using transfer learning as explained in [Torrey and Shavlik, 2010].

1. **Higher Start:** Even before fine tuning the model, the initial features are quick and more when we use transfer learning on the source model when compared to without transfer learning.
2. **Higher Slope:** The rate at which the features gets trained on the source model is steeper when compared to the non transfer learning model.
3. **Higher Asymptote:** The convergence factor of the transfer learning source model is better than the model without using transfer learning.

In short, a pre-trained model whose weights are being trained by someone else on a broad dataset and we "fine-tune" the algorithm with our own dataset. The pre-trained model will either provide the initialized weights resulting in a faster convergence or input as a fixed feature extractor for the interested action. For this study the experiments are based on fine tunning the features using pretrained network and changes are done as per the requirements. Figure 17, shows the transfer learning system architecture on basis of which experiments are performed.

The System architecture consists of 2 phases 'Training Phase' and 'Testing Phase'. Below are the process involved in each phase.

1. **Training Phase**
    - From the training dataset, individual images are taken as input and features are being extracted by converting the pixel matrix into vector format.
    - The features are pre-processed by using data augmentation techniques such as flipping, zooming, rotating, shifting the range, etc. and resulting reshaped image will be fit to 100x100 size and given to the system for further processing.
    - The pre-trained weights from ImageNet dataset is being used in order to calculate the weights of the input images, by imposing transfer learning on the neural network. Simultaneously extracted fruit labels are attached to those weights and system is trained as per these calculations.

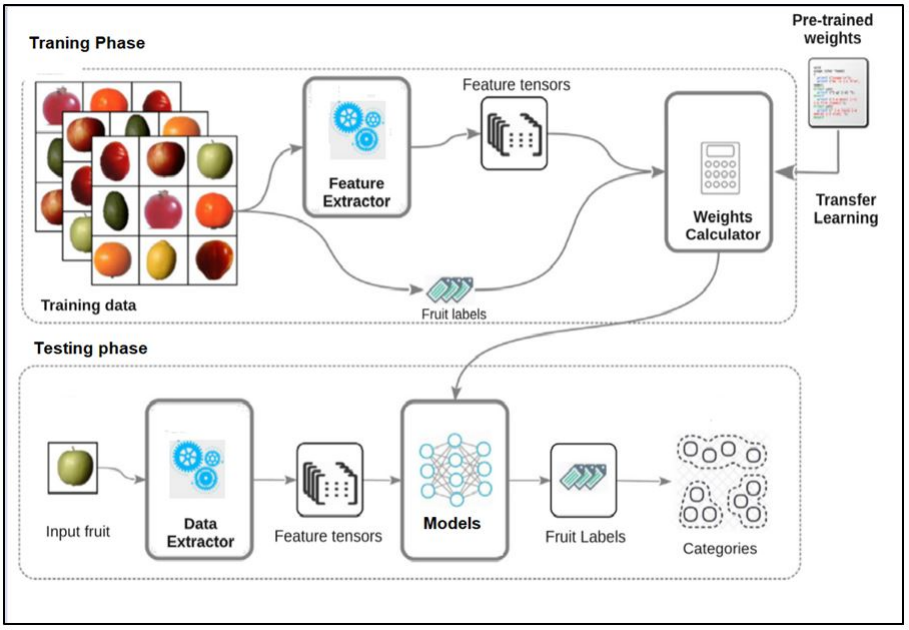

Figure 17: System Architecture for TL

- The calculated weights are fed to the network to fit transform and train the CCN architectural based models such as VGG, ResNet, DenseNet, etc.

2. **Testing Phase**
   - In order to test the fruit class, a fruit image is given to the system, where the same feature extraction process is executed.
   - The desired vector form of the image is fed to the trained model and by referring to the calculated weights each fruit class is appended with the score.
   - Finally, the label with maximum score is selected and classified as the predicted fruit class.

5 Model Implementation

As this study is based on series of experiments conducted on several convolutional neural network based architectures, the model implementation was very specific to individual architecture. ie., every model was trained and tested separately and their respective results were compared in order to decide the best transfer learning model for the given dataset.

5.1 Experiment Setup

This study is based on 2 phases of experiments, the first phase will be conducted using the smaller subset of Dataset with 21 classes and majority of models will be trained using this subset. The second phase will be conducted using larger Dataset of 120 classes a seen in figure 14. However below attributes are same for all the experiments such as:

- Hardware configuration of the system is 8 GB of Ram with Intel(R) Core(TM) i7-5600 CPU @2.60 GHz 2.59GHz of Processor.
- Experiment codes were executed on Jupyter Notebook.
- Keras libraries of version 2.2.4 were used and with Tensorflow version of 2.0.0.
- Few important libraries to be imported from keras.layers - Dense, Dropout, Flatten, Conv2D, MaxPool2D.
- From keras.preprocessing.image - ImageDataGenerator
- In order to evaluate the model, confusion matrix, classification report are imported from sklearn.metrics.
- To plot the graphs, pyplot is imported from matplotlib.

5.2 Experiment Phase 1

All the experiments under this phase has subset of dataset with following fruit classes- Apple Golden1, Avocado, Banana, Cauliflower, Cherry 1, Cocos, Kiwi, Lemon, Lychee, Mango, Onion Red, Orange,Pepper Green, Pineapple, Pomegranate, Potato White, Raspberry, Tamarillo, Strawberry, Tomato 4, Walnut.

5.2.1 Experiment 1- Self Designed CCN

The first experiment of this study was a self designed Convolutional Neural Network with Conv2D class, which creates 2D convolutional layers in keras. This 14 layer CNN is built independently from the very first step. Below are the process involved in building and training this model.

- Firstly, all the input images were extracted from the local drive folder.
- Input image were then converted into grey scale and resized the image from 100x100x3 to 100x100x1 as part of pre-processing techniques.
- The model was trained using Data Augmentation technique, which caused increase in the training data as data augmentation causes expansion in the size of the input data by modifying (rotating, flipping, zooming) the data.
- CNN model was built using “ReLU” and “Softmax” activation functions and “Adam” as optimizer.

1. ReLU - Rectified Linear Activation function is used as it does not saturate and very fast in response. The function returns 0 if it receives any negative input, but for any positive value xx it returns that value back. So it can be written as f(x)=max(0,x).

2. Softmax- The last layer of the network uses softmax as the activation function when dealing with multi class classification. SoftMax gives a discrete probability distribution over all the classes and hence results in only one class as output.

3. Adam- This optimizer calculates an exponential weighted moving average of the gradient and then squares it up. It is effective with noisy gradients and larger dataset. Optimizer is defined with Adam along with learning rate of 0.001 and the exponential decay rate for the first and second moment estimates as 0.9 and 0.999 respectively.

- The other factors which were important for training the CNN model were Epoch= 40, Steps per Epoch= 100 and Batch Size= 32.

Figure 18, shows the table of 14 layer CNN model configuration details.

Layer Name	Configuration
Input Layer	Size= 100x100x1
Conv2D Layer1	Filter Size= 8, Kernel Size= 5x5, Activation= ReLu
MaxPool Layer1	Pool Size= 2x2
Dropout Layer1	Rate= 0.25
Conv2D Layer2	Filter Size= 16, Kernel Size= 4x4, Activation= ReLu
MaxPool Layer2	Pool Size= 2x2
Dropout Layer2	Rate= 0.25
Conv2D Layer3	Filter Size= 32, Kernel Size= 4x4, Activation= ReLu
MaxPool Layer3	Pool Size= 2x2
Dropout Layer3	Rate= 0.25
Flatten Layer1	flatten the input tensor
Dense Layer1	Units= 512, Activation= ReLu
Dropout Layer4	Rate= 0.5
Dense Layer2	Units= 21, Activation= Softmax
Output Layer	Size= 1x21

Figure 18: Self-Designed CNN Configuration Details

5.2.2 Experiment 2- VGG16 using Transfer Learning

All the experiments which are conducted using transfer learning approach, uses same hyper parameters and pre-processing techniques. Few important steps executed to train the model using transfer learning of CNN's architecture pre-trained models are as below.

- It is very much important to import the model libraries from "Keras. application" which is very specific to the model used for training.

- In order to pre-process the new dataset, with the pretrained weights of ImageNet, specific library is required to be imported for specific model configuration. For exam-

ple when training the VGG16 model it should be used as from "keras.applications.vgg16 import preprocess input".

- As the default input size of the ImageNet images is 224x224x3, we are allowed to modify this size as per the requirement. Thereby, we modify the size as 100x100x3 as per the experimental requirements.
- While defining the specific model, along with the input shape, weight argument should be defined and set as "imagenet".
- The hyperparameters are kept same for all the transfer learning based experiments; such as Epoch = 5, Batch Size= 32, Optimizer function as rmsprop and Loss function as categorical crossentropy (as it is multi-class classification).

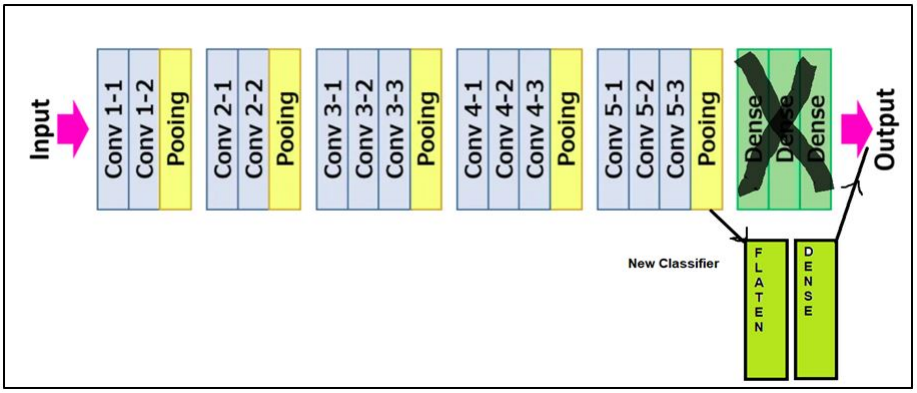

Figure 19: Layers of VGG16 Using Transfer Learning

Figure 19, shows the architectural block diagram of the convolutional layers configuration for VGG16 using transfer learning. As seen in the figure, VGG16 has 16 layers in combination with 2 or 3 Convolution layer and a max-pooling layer. The initial layers are frozen layers and we cannot modify them. However, in order to perform transfer learning using this convolution architecture, the bottom layers are modified and replaced with new classifier layer.

- A Flatten layer to change the vector shape of data into a understandable format for the Dense layer.
- The final Dense layer is passed with "Softmax" activation function and the classification size to be 21 (as per the total number of classes).

5.2.3 Experiment 3- ResNet50 using Transfer Learning

As discussed in the above section, any experiment conducted using transfer learning will have same hyper-parameters. The fine-tuning and pre-processing steps will also remain same. However, the keras libraries which has to be imported will keep changing as per the architecture. For example for training using ResNet50, from keras.applications ResNet50 has to be imported. Similarly the pre-processing library would be from keras.applications.resnet50 import preprocessing input.

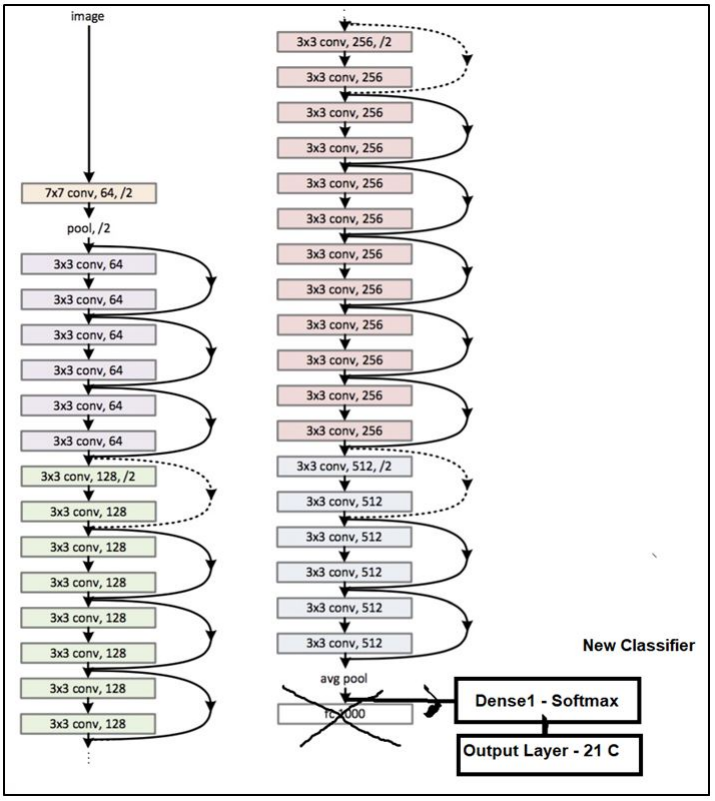

Figure 20: Layers of ResNet50 Using Transfer Learning

Figure 20, shows the architectural block diagram of the convolutional layers configuration for ResNet50 using transfer learning. Resnet is combination of Identity and convolutional blocks. As seen ResNet50, is in divided in 4 stages, with residual blocks containing 3 convolutional layers each. For this experiment initial layer tenability is disabled as ResNet50 neural network has batch-normalization layers and using the pre-trained model causes issues with BN layers, if the target dataset on which model is being trained on is different from the originally used training dataset. This is because the BN layer would be using statistics of training data, instead of one used for inference. The last layer Fully Connected layer of the architecture is replaced with the new classifier of Dense Layer along with Softmax activation function and an output layer to classify 21 class outputs.

5.2.4 Experiment 4, 5, 6, 7 & 8 -Using Transfer Learning

Series of experiments were conducted on different convolution neural network architectures using transfer learning as given below.

- MobileNet
- InceptionV3
- Desnet121
- InceptionResNetV2
- Xception

The hyper-parameters and the procedure used for training the above architecture models

are same as of VGG16 and ResNet50. However, the results of VGG16 and ResNet50 are better than the above list.

5.3 Experiment Phase 2

As the main motive of this study is to investigate the best model using transfer learning on CNN architecture. And, as seen in figure 14, the module design. The top 2 models from experiment phase 1, will be trained and evaluated with the larger dataset. The whole set up and system configuration remains the same just increase in factors, class size = 120, training image= 60498, testing image=20622.

Under the phase 2, experiments are conducted on VGG16 and ResNet50.

6 Model Evaluation and Visualization

To draw any conclusion from the experiments conducted it is very much important to have a combination of model evaluation and visualization. In order to evaluate any model better visualisation techniques and tools are to be used. Thereby, in this study visualization is performed on both the model level as well as data level. Graphical plots are used to analyse the model's accuracy and loss using python code. Similarly in order to analyze the prediction of classes "Confusion Matrix" and "Classification Report" is used on individual models. Further, all the results of the individual model is accumulated, in an excel file so that visualization on the complete result set can be performed using "Tableau" tool.

Below are the factors used to analyze the model's efficiency in this study:

1. Training and Testing Accuracy and Loss- During the training and validation stage, accuracy shows the model's performance in an easy way. Loss values shows how bad or good a model's response after every iteration or epoch of optimization.
2. Precision- For a given class, precision is the number of true positives divided by the number of true positives plus the number of false positives.
3. Recall- Recall calculates how many of the actual positives the model capture through labeling it as positive (true positive).
4. F1 Score- The F1 score is the harmonic mean of precision and recall in order to push the extreme values.
5. Training Time- This factor is very negligible while comparing the model, it shows how dense the layers of the architecture is and time efficiency.

All the experiments in phase 1 are evaluated based on above factors and below are the results as tabulated in figure 21.

Model Name	Training Accuracy	Testing Accuracy	Training Time	Precision	Recall	F1
VGG16	99.90%	99.80%	01:36:16	0.998	0.998	0.998
ResNet50	99.80%	89%	01:30:03	0.948	0.914	0.914
MobileNet	99%	69%	00:28:59	0.782	0.704	0.657
InceptionV3	79%	53%	00:53:57	0.627	0.529	0.493
Desnet121	99%	88%	02:04:30	0.93	0.876	0.874
InceptionResNetV2	90.73%	61.05%	01:28:00	0.694	0.607	0.577
Xception	96%	60%	01:38:35	0.772	0.592	0.552
CNN-Conv2D	99%	99%	01:07:30	0.928	0.979	0.979

Figure 21: Experiment Results Table for 21 Class

6.1 Evaluation Techniques

Two techniques are used to evaluate; Plot of accuracy and loss versus epoch values and Plot of confusion matrix.

6.1.1 Accuracy and Loss V/S Epochs

Upon successful completion of model training, accuracy and loss values change with every epoch increment. There by, to understand the trend between accuracy and loss in training set as well as validation (testing) set with respect to epoch values, graphs are been plot by using the ".history" feature of keras model. Graphs are as shown in figure 22 and 23. This graphs are very important in order to further fine tune the model, provided it helps understand which epoch gives the best accuracy and loss. However, by using callback functions the model can stop at the last best reached accuracy and loss values.

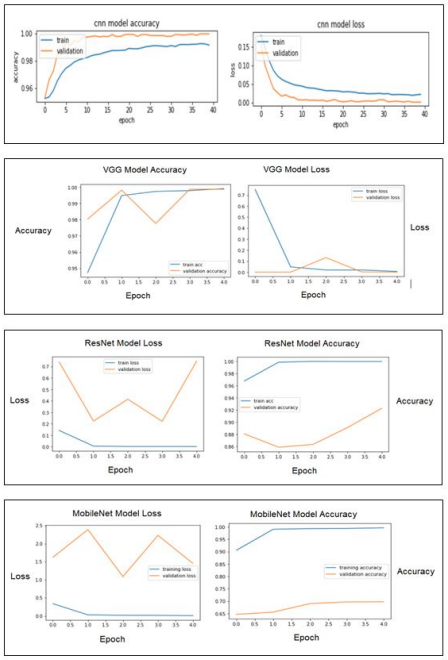

Figure 22: Accuracy & Loss V/s Epoch for Experiments 1,2,3,4

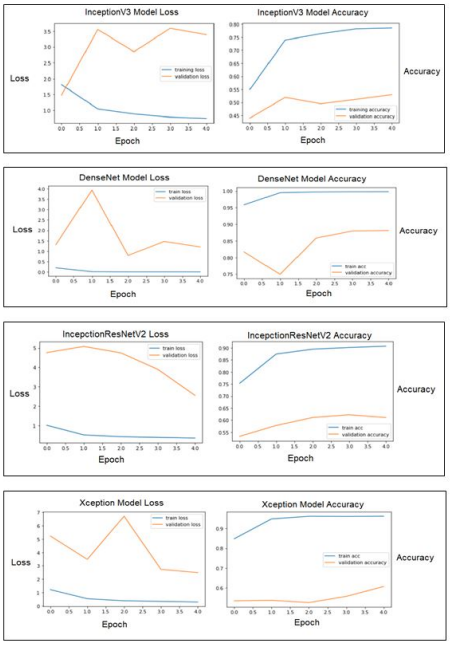

Figure 23: Accuracy & Loss V/s Epoch for Experiments 5,6,7& 8

6.1.2 Confusion Matrix

Displaying the predicted fruits and its label, when dealing with huge number of data is not a accurate and feasible evaluation method. Instead, the best way is to analyse how many fruits or classes are wrongly predicted and how many are the truly predicted from the trained model. In order to achieve this a plot of "Confusion Matrix" is required which gives the correct count of true predictions and false predictions. As the experiments are based on multi-class classification displaying and analysing confusion matrix is not feasible. There by, the alternative is to analyze the Classification Report provided by the sklearn.metrics library. This report gives the precision, recall and F1 score values for individual predicted class as seen in figure 24, and the highlighted data is the average precision, recall and F1 score value, which will be used to compare with other models. Figure 24 shows the classification report for VGG16, likewise, same plots have been analysed for other models as well.

	precision	recall	f1-score	support
Apple Golden 1	0.965	1.000	0.982	164
Avocado	1.000	1.000	1.000	143
Banana	0.994	1.000	0.997	166
Cauliflower	1.000	1.000	1.000	234
Cherry 1	1.000	1.000	1.000	164
Cocos	1.000	1.000	1.000	166
Kiwi	1.000	1.000	1.000	156
Lemon	1.000	0.994	0.997	164
Lychee	1.000	1.000	1.000	166
Mango	1.000	1.000	1.000	166
Onion Red	1.000	1.000	1.000	150
Orange	1.000	1.000	1.000	160
Pepper Green	1.000	1.000	1.000	148
Pineapple	1.000	1.000	1.000	166
Pomegranate	1.000	1.000	1.000	164
Potato White	1.000	0.960	0.980	150
Raspberry	1.000	1.000	1.000	166
Strawberry	1.000	1.000	1.000	164
Tamarillo	1.000	1.000	1.000	166
Tomato 4	1.000	1.000	1.000	160
Walnut	1.000	1.000	1.000	249
accuracy			0.998	3532
macro avg	0.998	0.998	0.998	3532
weighted avg	0.998	0.998	0.998	3532

Figure 24: Classification Report of VGG16

6.2 Visualization Techniques

Two techniques are used to visualize the results; A bar chart analysis and tableau reports.

6.2.1 Bar Chart Analysis

Based on the over all results of all the models as shown in table of figure 21, a dual axis bar chart is plotted which helps to analyze the results and gives the best model for experiment phase1. The decision is made by comparing Training Accuracy and Testing Accuracy, along with the training time.

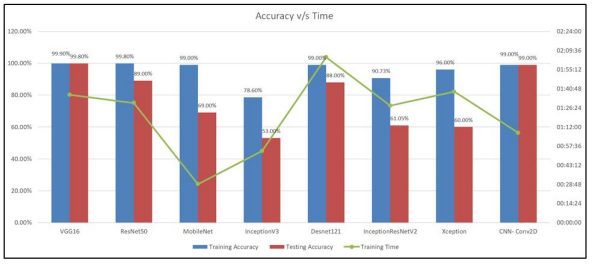

Figure 25: Accuracy v/s Time Bar Chart

The Figure 25, shows the accuracy v/s time bar chart, which is a very useful to decide the top 2 best models for the this study. Provided many useful conclusions are also drawn from this chart, which is discussed in further section.

6.2.2 Tableau Report Analysis

In order to analyse the collective results of classification reports from all the models a specialised tool is used. Tableau is a very well known visualization tool majorly used for larger data. A “Heatmap” and a “Bubble Chart” is produced from the collated data on individual fruit class, so that analysis can be performed easily. Figure 26, shows the heatmap for individual fruit & vegetable class F1 Score of all the models. The darker shades show the best F1 score and lighter shade show the lower F1 score. Figure 27, shows the 4 quadrant bubble chart of Precision versus Recall of individual fruit classes

with its respective model. The bubble size here shows that the intensity of precision and recall. Bigger the bubble size, better the precision and recall value.

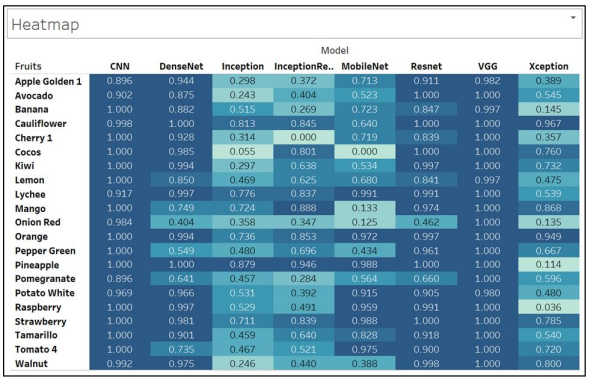

Figure 26: Model v/s Class F1 score Heatmap

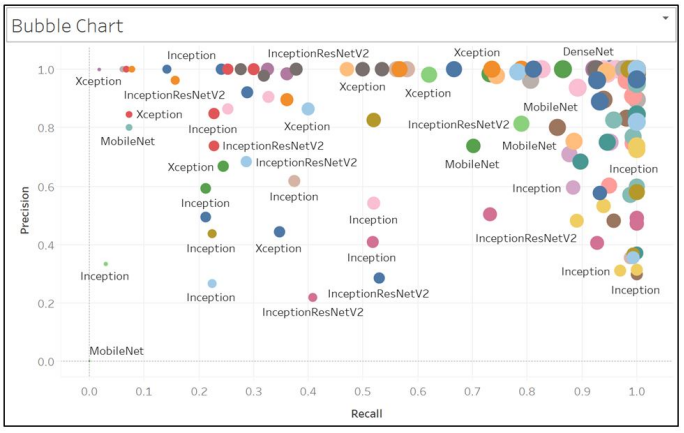

Figure 27: Class Precision V/s Class Recall

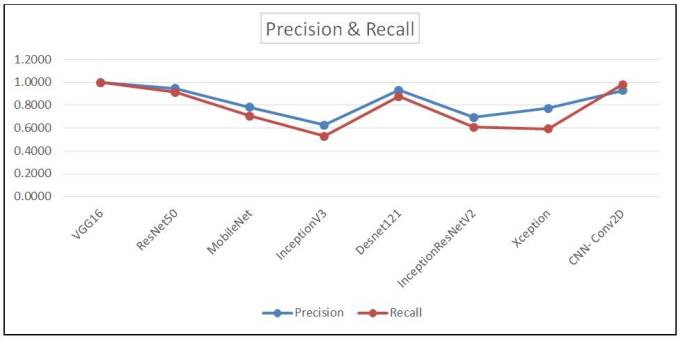

Figure 28: Trend of Precision & Recall

6.3 Evaluation for Phase2 Experiments

As per the results of above visualizations, the top 2 transfer learning models for the given dataset are VGG16 and ResNet50. However, the training time is not considered majorly, as it is the least factor for evaluation because any model is trained only once and due to the complex colvolutional layer structures in VGG and ResNet, as discussed in the

detailed architecture of these models, the training time will be more compared to others. Based on the "Testing Accuracy", "Precision", "Recall" and "F1 score" the top 2 models are selected. Same evaluation techniques are used to evaluate the experiments in phase 2, on VGG16 and ResNet50 with 120 classes. The results are as tabulated below;

Model Name	Training Accuracy	Testing Accuracy	Training Time	Precision	Recall	F1
VGG16 (120 class)	99%	95.30%	09:58:45	0.958	0.952	0.951
ResNet50 (120 class)	99.35%	54.40%	08:36:46	0.774	0.538	0.536

Figure 29: Experiment Phase2 Results

The figure 30, shows the accuracy and loss trends per epochs for both VGG16 and ResNet50;

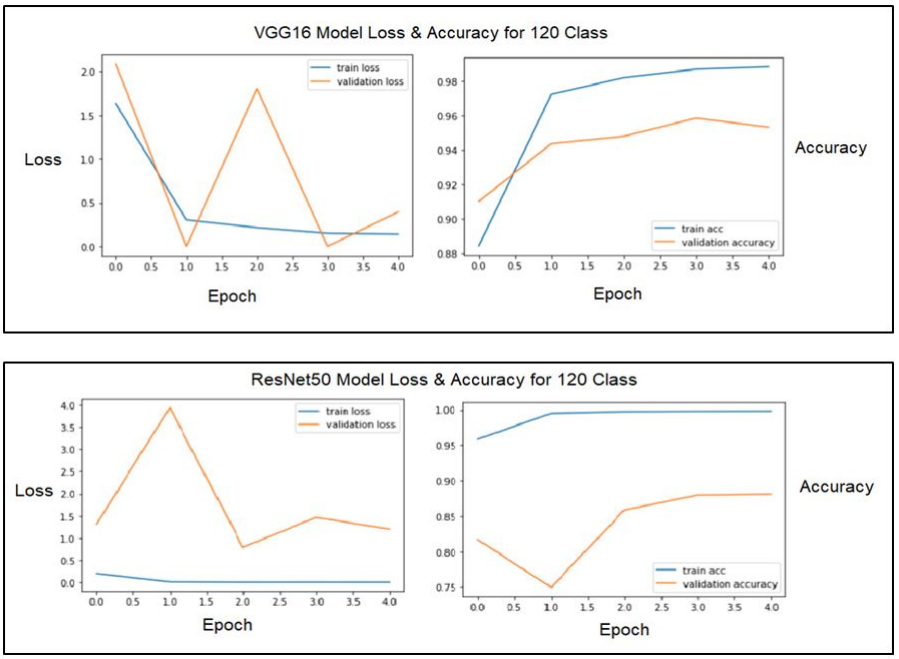

Figure 30: Accuracy & Loss V/s Epoch of Phase2 Experiment

7 Conclusion and Future Work

Based on the above results from figure 28, we can accept the hypothesis, that when compared to the other 7 different and advanced types of transfer learning based convolutional neural network architectures, VGG16 is the most efficient and effective architectural model for applying transfer learning. Based on the series of experiments being performed, we can now answer the research questions stated in the beginning of this study.

- While comparing the transfer learning based models with a self designed CNN, it is proved from the figure 21, 25 and 26 that a self designed 14 layer CNN is more efficient and accurate from all the other transfer learning models. However, VGG16 has slittly better accuracy and F1 score than CNN, but CNN beats VGG16 when it comes to training time.
- As Stated the VGG16 has 0.8% more accuracy and from figure 28, analysing the trend of precision and recall, we can say that VGG16 is better than self designed CNN, as it is well balanced and consistent in both predicting the true positives and true negatives.
- Every model architecture have there own pros and cons. While performing these experiments, respective model's qualities are evaluated. Below are the important outcomes as seen from the bar chart of figure 25:
    1. VGG16 is more accurate than older versions of CNN like, Xception, GoogleNet (InceptionV3) model.
    2. MobileNet takes the shortest time to train, as its whole purpose was to make the architecture compact and simple.
    3. DenseNet is the most densely layered architecture, there by making it more complex. Thus, we see in our results the maximum time taken to train the model is by DenseNet.
    4. InceptionResNetV2 was structured to include both the benefits of Inception and ResNet. However, our results doesn't prove so. In fact, it is the lowest performing model followed by InceptionV3.
- By analysing the heatmap and the bubble chart, it can be said that there are some common trends seen in all the models like; Orange fruit class is the best predicted by all the models. And, fruit class like Cocos, Cherry and Walnut are predicted false by most of the models.

Overall, by performing these studies we can conclude that the base architecture of convolutional neural network is the best for image classification. However, further more experiments should be performed and tested on the same dataset with more image classes because as seen few fruit classes are false predicted by most of the models, which shows that the dataset is slightly imbalanced and requires more input images with strong features so that the model can be better trained. In order to understand better and prove the hypothesis right for this study, more experiments should have been performed by fine-tuning the existing models and self designing VGG and ResNet architectures. However, the future scope of this study is to perform these fine tuning of models and enhance this study further, by creating a real time application which can be used in the supermarkets to identify the fruit or vegetable and assign the price accordingly. Thereby, avoiding the manual process by automating the system.

References

[Adigun et al.,] Adigun, J., Okikiola, F., Aigbokhan, E., and Rufai, M. Automated system for grading apples using convolutional neural network.

[Bargoti and Underwood, 2017a] Bargoti, S. and Underwood, J. (2017a). Deep fruit detection in orchards. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3626–3633. IEEE.

[Bargoti and Underwood, 2017b] Bargoti, S. and Underwood, J. P. (2017b). Image segmentation for fruit detection and yield estimation in apple orchards. *Journal of Field Robotics*, 34(6):1039–1060.

[Barnett, 2015] Barnett, A. (2015). The nature of crops: Why do we eat so few of the edible plants?

[Bolle et al., 1996] Bolle, R. M., Connell, J. H., Haas, N., Mohan, R., and Taubin, G. (1996). Veggievision: A produce recognition system. In *Proceedings Third IEEE Workshop on Applications of Computer Vision. WACV'96*, pages 244–251. IEEE.

[Cheng et al., 2017] Cheng, H., Damerow, L., Sun, Y., and Blanke, M. (2017). Early yield prediction using image analysis of apple fruit and tree canopy features with neural networks. *Journal of Imaging*, 3(1):6.

[Duong et al., 2020] Duong, L. T., Nguyen, P. T., Di Sipio, C., and Di Ruscio, D. (2020). Automated fruit recognition using efficientnet and mixnet. *Computers and Electronics in Agriculture*, 171:105326.

[Everingham et al., 2015] Everingham, M., Eslami, S. A., Van Gool, L., Williams, C. K., Winn, J., and Zisserman, A. (2015). The pascal visual object classes challenge: A retrospective. *International journal of computer vision*, 111(1):98–136.

[Femling et al., 2018] Femling, F., Olsson, A., and Alonso-Fernandez, F. (2018). Fruit and vegetable identification using machine learning for retail applications. In *2018 14th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)*, pages 9–15. IEEE.

[Fruitsinfo.com, 2020] Fruitsinfo.com (2020). Alphabetical list of fruits.

[He et al., 2016] He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

[Hossain et al., 2018] Hossain, M. S., Al-Hammadi, M., and Muhammad, G. (2018). Automatic fruit classification using deep learning for industrial applications. *IEEE Transactions on Industrial Informatics*, 15(2):1027–1034.

[Howard et al., 2017] Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., and Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*.

[Huang et al., 2017] Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K. Q. (2017). Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708.

[Huang et al., 2019] Huang, Z., Cao, Y., and Wang, T. (2019). Transfer learning with efficient convolutional neural networks for fruit recognition. In *2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, pages 358–362. IEEE.

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

[Jeong and Yoe, 2018] Jeong, S. and Yoe, H. (2018). Design of deep learning based fruit classification system platform. In *Proceedings of the International Conference on Wireless Networks (ICWN)*, pages 64–67. The Steering Committee of The World Congress in Computer Science, Computer ....

[Kamilaris and Prenafeta-Boldú, 2018] Kamilaris, A. and Prenafeta-Boldú, F. X. (2018). Deep learning in agriculture: A survey. *Computers and electronics in agriculture*, 147:70–90.

[Keras, 2020] Keras (2020). Keras keras application.

[Khan and Debnath, 2019] Khan, R. and Debnath, R. (2019). Multi class fruit classification using efficient object detection and recognition techniques. *International Journal of Image, Graphics and Signal Processing*, 11(8):1.

[Koirala et al., 2019] Koirala, A., Walsh, K. B., Wang, Z., and McCarthy, C. (2019). Deep learning–method overview and review of use for fruit detection and yield estimation. *Computers and Electronics in Agriculture*, 162:219–234.

[Krizhevsky et al., 2012] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.

[Larada et al., 2018] Larada, J. I., Pojas, G. J., Ferrer, L. V. V., et al. (2018). Postharvest classification of banana (musa acuminata) using tier-based machine learning. *Postharvest biology and technology*, 145:93–100.

[Mureşan and Oltean, 2018] Mureşan, H. and Oltean, M. (2018). Fruit recognition from images using deep learning. *Acta Universitatis Sapientiae, Informatica*, 10(1):26–42.

[Nikhitha et al., 2019] Nikhitha, M. et al. (2019). Fruit recognition and grade of disease detection using inception v3 model. In *2019 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA)*, pages 1040–1043. IEEE.

[Oltean, 2017] Oltean, M. (2017). Fruits 360 a dataset of images containing fruits and vegetables.

[Puttemans et al., 2016] Puttemans, S., Vanbrabant, Y., Tits, L., and Goedemé, T. (2016). Automated visual fruit detection for harvest estimation and robotic harvesting. In *2016 Sixth International Conference on Image Processing Theory, Tools and Applications (IPTA)*, pages 1–6. IEEE.

[Rizwan, 2018] Rizwan, M. (2018). Vgg16 – implementation using keras.

[Rocha et al., 2010] Rocha, A., Hauagge, D. C., Wainer, J., and Goldenstein, S. (2010). Automatic fruit and vegetable classification from images. *Computers and Electronics in Agriculture*, 70(1):96–104.

[Russakovsky et al., 2015a] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. (2015a). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252.

[Russakovsky et al., 2015b] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. (2015b). Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252.

[Sa et al., 2016] Sa, I., Ge, Z., Dayoub, F., Upcroft, B., Perez, T., and McCool, C. (2016). Deepfruits: A fruit detection system using deep neural networks. *Sensors*, 16(8):1222.

[Saha, 2018] Saha, S. (2018). A comprehensive guide to convolutional neural networks — the eli5 way.

[Semary et al., 2015] Semary, N. A., Tharwat, A., Elhariri, E., and Hassanien, A. E. (2015). Fruit-based tomato grading system using features fusion and support vector machine. In *Intelligent Systems' 2014*, pages 401–410. Springer.

[Shearer, 2000] Shearer, C. (2000). The crisp-dm model: the new blueprint for data mining. *Journal of data warehousing*, 5(4):13–22.

[Siddiqi, 2019] Siddiqi, R. (2019). Effectiveness of transfer learning and fine tuning in automated fruit image classification. In *Proceedings of the 2019 3rd International Conference on Deep Learning Technologies*, pages 91–100.

[Svozil et al., 1997] Svozil, D., Kvasnicka, V., and Pospichal, J. (1997). Introduction to multi-layer feed-forward neural networks. *Chemometrics and intelligent laboratory systems*, 39(1):43–62.

[Szegedy et al., 2015] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9.

[to Learn, 1995] to Learn, N. L. (1995). Learning to learn: Knowledge consolidation and transfer in inductive systems.

[Torrey and Shavlik, 2010] Torrey, L. and Shavlik, J. (2010). Transfer learning. In *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques*, pages 242–264. IGI Global.

[Tóth et al., 2016] Tóth, B. P., Tóth, M. J., Papp, D., and Szücs, G. (2016). Deep learning and svm classification for plant recognition in content-based large scale image retrieval. In *CLEF (Working Notes)*, pages 569–578.

[Xiang et al., 2019] Xiang, Q., Wang, X., Li, R., Zhang, G., Lai, J., and Hu, Q. (2019). Fruit image classification based on mobilenetv2 with transfer learning technique. In *Proceedings of the 3rd International Conference on Computer Science and Application Engineering*, pages 1–7.

[Xiong et al., 2018] Xiong, J., Liu, Z., Lin, R., Bu, R., He, Z., Yang, Z., and Liang, C. (2018). Green grape detection and picking-point calculation in a night-time natural environment using a charge-coupled device (ccd) vision sensor with artificial illumination. *Sensors*, 18(4):969.

[Yamamoto et al., 2014] Yamamoto, K., Guo, W., Yoshioka, Y., and Ninomiya, S. (2014). On plant detection of intact tomato fruits using image analysis and machine learning methods. *Sensors*, 14(7):12191–12206.

[Yu et al., 2017] Yu, S., Jia, S., and Xu, C. (2017). Convolutional neural networks for hyperspectral image classification. *Neurocomputing*, 219:88–98.

[Zhang et al., 2014] Zhang, Y., Wang, S., Ji, G., and Phillips, P. (2014). Fruit classification using computer vision and feedforward neural network. *Journal of Food Engineering*, 143:167–177.

[Zhang and Wu, 2012] Zhang, Y. and Wu, L. (2012). Classification of fruits using computer vision and a multiclass support vector machine. *sensors*, 12(9):12489–12505.

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