Pune Institute of Computer Technology
Dhankawadi, Pune

A SEMINAR REPORT
ON

Rot detection in Fruits Using Deep Learning

SUBMITTED BY

Name : Krishiv Dakwala
Roll No. : 33216
Class: TE-10

Under the guidance of
Dr. Anant M.Bagade

DEPARTMENT OF INFORMATION TECHNOLOGY
Academic Year 2021-22

DEPARTMENT OF INFORMATION TECHNOLOGY
Pune Institute of Computer Technology
Dhankawadi, Pune-43

CERTIFICATE

This is to certify that the Seminar report entitled

"ROT DETECTION IN FRUITS USING DEEP LEARNING"

Submitted by
Krishiv Dakwala Roll No. : 33216

is a record of bonafide work carried out by him/her under the supervision and guidance of Prof. Dr. Anant M.Bagade in partial fulfillment of the requirement for TE (Information Technology Engineering) – 2019 course of Savitribai Phule Pune University, Pune in the academic year 2021-22.

Dr. Anant M.Bagade	Dr. A.M.Bagade
Guide	Head of Department

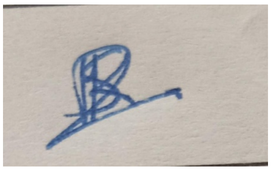
Place:Pune
Date: 14/11/2021

ACKNOWLEDGEMENT

It is my pleasure to present report on ”Rot Detection in Fruits Using Deep Learning”. First of all, I would like to thank our Seminar Coordinator Prof. Sachin Shelke, Head of Department Dr. A.M Bagade and Principal R.Shreemati for this opportunity.

I would also genuinely express my gratitude to my guide Dr. Anant M.Bagade, Department of Information Technology for his constant guidance and help. He has constantly supported me and has played crucial role in completion of this report.His motivation and encouragement from beginning till end to make this seminar a success.

Last but not the least I would thank all the faculty, my parents and friends who have helped me.



(Krishiv Dakwala)

Contents

List of Tables

List of Figures

Abstract

In recent years, Artificial intelligence has been at the crux of many advancements across different industries such eCommerce, Finance and Healthcare. The introduction of new Machine learning techniques in the agri-food industry could become a turning point. The quality of the produce is one of the major factors that drives this industry. It plays an important role in consumer consumption and thereby affecting its sales. For a country like India, which heavily depends on agriculture, introduction of modern methods in agriculture is vital for increasing the efficiency and thus increasing the overall produce. AI based autonomous systems could cut down the manual labor which will hence reduce processing times and can have a huge impact on sales. Deep learning along with computer vision can be used to analyze the quality of the fruits and vegetables. It can also be used to grade the goods according to their freshness and can even identify rots in fruits and vegetables. Traditionally, manual labor is used to perform the above-mentioned tasks, but with the introduction of the proposed technology, we can cut down the costs and reduce the processing time.

1 INTRODUCTION

Fruit spoiling has a huge economic impact; it is estimated that roughly a third of the cost of fruit is spent on rotting materials. Furthermore, the sale of fruits will be impacted because consumers believe that spoiled fruits are harmful to their health as reduced concentrations of amino acids, vitamins, sugar/glucose, and other nutrients inevitably raise public concerns about edibility issues, prompting discussions on how to prevent or slow the decaying process.

Rot detection in fruits is significant because of the importance of food status in people's lives and contribution to the economy, yet the manual operation is time-consuming. Automation of grading through the use of computerized methods is thought to be the answer to this problem.

When fruit deteriorates, established research evidence demonstrates that it goes through a sequence of metabolic transformations that result in changes in its physical conditions, such as color and shape. The majority of these characteristics can be recorded. The most cost-effective solution is projected to be a computer vision-based technique. In this paper, I propose an efficient and effective machine vision system based on cutting-edge deep learning techniques, as well as a cost-effective solution for automating the rot detection process in fruits.

To discover the best model for detecting rotten produce, I trained, tested, and compared the performance of several deep learning models, including ResNet, DenseNet, Xception, VggNet, and NASNet.

2 MOTIVATION

E-commerce has exploded in popularity over the last few years. People are increasingly purchasing goods online, including their daily need for fruits and vegetables. The issue here is that the consumer is unaware of the quality of the fruits before purchasing them. Consumers may receive rotten or old fruits, resulting in food and money waste. Consumers will receive the highest quality items if automated technologies can replace manual segregation and removal of rotting food. This can boost sales and cut down on fruit waste. Segregation right after harvesting the vegetables might be a better way to handle this problem. This could cut total waste as well as the time and effort required to manually separate rotten produce. This can save expenses, improve quality, and raise revenues overall.

3 AIM AND OBJECTIVES

3.1 Aim

The aim of this seminar is to study various algorithms and identify the best classification model which is needed for automating the process of segregating rotten fruits.

3.2 Objectives

To build and compare several deep learning models and discover the best model for the task.

4 LITERATURE SURVEY

4.1 Introduction

Similar work has been done in this field and different deep learning models are developed and various architectures are used for quality analysis of fruits and vegetables. Various quality issues include rots, diseases, ageing etc.

4.2 A Survey on Papers

The Following table shows the literature survey by comparing techniques propose in various references:

Sr.	Authors	Summary	Limitations
1	Nazrul Ismail, Owais A. Malik, 2020 [1]	Reviewed methodologies that perform automatic visual inspection of fruits' freshness and appearance. Compared various deep learning models with respect to accuracy.	The speed of the model is not taken into account Some modes such as VggNet and Xception are not taken into study
2	Yuhang Fu[2]	Reviewed various architectures such as AlexNet, VggNet, ResNet, Inception, Google Net, etc. Proposed a grading system for fruit and vegetable freshness	Some of the newer versions of the architectures (such as Vgg19, ResNet50, ResNet v2 etc.) were not reviewed. Xception was not considered for the review. The Dataset was small considering the no. of classifications made.
3	Richa Shah, Pooja Gujarathi, Parth Unadkat, Rijoosinh Mulik, A. N. Bandal [3]	Proposed a clustering-based framework which aims to predict the quality of fruits and vegetables	One of the major drawbacks of the proposed model is the use of ANN instead of CNN which reduces the performance speed greatly.

4	Mrs. Pushpa B R, Ms. Tripulla K H and Ms. Meghana T K [4]	Proposed and evaluated an image processing based approach to detect fruit disease.	Features are directly extracted using image processing which leaves no room for deep learning models
5	José Naranjo-Torres, Marco Mora, Ruber Hernández-García, Ricardo J. Barrientos, Claudio Fredes, Andres Valenzuela [5]	Reviewed the applications of CNN in Fruit image processing Reviewed various CNN architectures – AlexNet, Vgg16, MobileNet, Inception V3, ResNet 50.	Used a single fruit dataset for quality classification (Apple-NDDA) dataset Some of the newer versions of the architectures (Vgg19, ResNet v2) were not taken into consideration. Xception architecture was not taken for review.

Table 1: Survey on Papers

5 METHODOLOGIES AND TERMINOLOGIES

5.1 Technologies Used

5.1.1 Deep Learning

Deep learning is a machine learning and artificial intelligence (AI) technique that simulates how humans learn. In statistics and predictive modelling, deep learning is a critical component. Data scientists who are responsible for gathering, analyzing, and interpreting massive amounts of data will find deep learning to be incredibly useful because it speeds up and simplifies the process.

At its most basic level, deep learning may be thought of as a way to automate predictive analytics. Unlike traditional machine learning algorithms, which are linear, deep learning algorithms are designed in a hierarchy of increasing complexity and abstraction.

5.1.2 CNN

The Convolutional Neural Network (CNN) is a Deep Learning algorithm that takes an input image and assigns relevance (learnable weights and biases) to various components of the image, allowing for better image discrimination.

Image categorization is one of the most often used uses of this architecture. Several convolutional layers, as well as nonlinear and pooling layers, make up the neural network. When an image is processed through one convolution layer, the output of the first layer becomes the input of the second layer. For each subsequent layer, the technique is repeated. A fully connected layer must be introduced after a series of convolutional, non-linear, and pooling layers. This layer receives the output data from convolutional networks. When you connect a fully connected layer to the network's end, you receive an N-dimensional vector, where N is the number of classes the model can choose from. After a series of convolutional, non-linear, and pooling layers, a fully connected layer must be added. The output information from convolutional networks is fed into this layer. When a completely connected, layer is attached to the network's end, it produces an N-dimensional vector, where N is the number of classes from which the model chooses the appropriate one.

5.1.3 Transfer Learning

Transfer learning is a machine learning technique in which a model created for one job is utilized as the basis for a model on a different task.

Given the vast compute and time resources required to develop neural network models on these problems, as well as the huge jumps in skill that they provide on related problems, it is a popular approach in deep learning where pre-trained models are used as the starting point on computer vision and natural language processing tasks.

5.2 Process Diagram

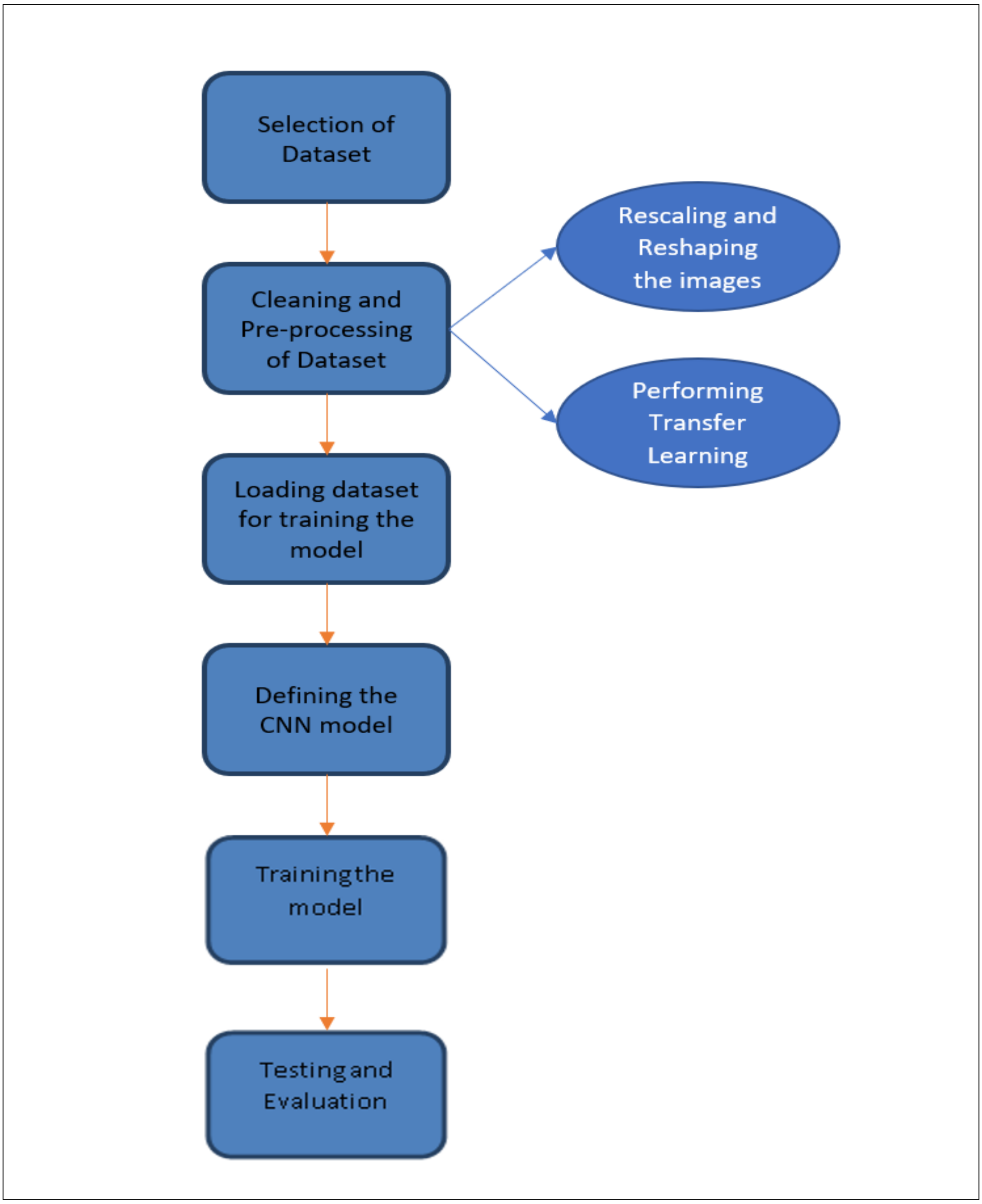


Figure 1: Process Diagram

5.3 Methodology

As shown in the process diagram the methodology followed to achieve the desired task is as follows:

Step 1: Selection of dataset

The dataset selected for the training purpose can be found here - https://www.kaggle.com/srin fresh-and-rotten-for-classification.
The dataset contains:

- Images of rotten apples, bananas, and oranges
- Images of fresh apples, bananas, and oranges
- There are a total of 5501 training images
- There are a total of 1384 testing images
- This dataset is already cleaned and ready for further processing

Step 2: Cleaning and Pre-processing

The dataset selected is already cleaned and ready for further processing. For further processing we will perform data augmentation to feed the data in our CNN. Processes included in data augmentation are –

- First, we will use transfer learning. Transfer learning is reusing a pre-trained model on a new problem, with the model leveraging knowledge from a previous task to increase generalization on a subsequent job. We will be using the following architectures –
  - VggNet -VggNet16, VggNet19
  - ResNet – ResNet, ResNet50, ResNet v2
  - NasNet
  - Xception
  - DenseNet
- Next, we must rescale our image to 1/255, which normalizes and transforms each pixel in the range of 0,255 to 0,1. Because different images have different pixel ranges, we do this to treat all of them the same way.
- Now, A horizontal and vertical flip will be applied to our train batches, but not to our testing data. We don't want to supplement our testing data; instead, we want to make sure it's brand fresh.
- Finally, we will reshape the data.
- This dataset is already cleaned and ready for further processing

Step 3: Loading the dataset

This is a very simple step where all we have to do is create the variables for our test And train batches and set the path for the same.

Step 4: Defining the CNN model

For constructing the convolutional neural network, we will add the following layers: We will add a convolutional layer as our first layer. It will have 32 filters with a filter size of (3, 3). We will use relu as the activation function. After this we will add a max pooling layer with a pool size of (2, 2). Now we will need to add another Conv2D layer but this time having 64 filters with a size of (3,3), continuing with the relu activation. We will also add another MaxPooling2D layer that will have a pool size of 2, 2. Now, we'll create our dense layers, but first, we'll use the Flatten function to flatten our input. We'll add two dense layers with 128 neurons each. For these layers, we will continue to use the relu activation function. The final layer will contain 6 neurons for 6 categories (fresh apples, fresh bananas, fresh oranges, rotten apples, rotten bananas, rotten oranges). This layer will have an activation of softmax.

Step 5: Training the model

For compiling and training our model we will use adam as our optimizer and categorical cross entropy for our loss function. Because we have multi-class classification problems, we apply this loss function. We use this function since our images can be classified into a number of different categories.

Step 6: Training the model

For compiling and training our model we will use adam as our optimizer and categorical cross entropy for our loss function. Because we have multi-class classification problems, we apply this loss function. We use this function since our images can be classified into a number of different categories.

Step 7: Testing and evaluation

Finally, after the model is trained, we will test the model on the test batches to calculate model metrics such as speed and accuracy. Note that, we will use these 6 steps to iterate over all the chosen architectures (Vgg16, Vgg19, ResNet, ResNet50, ResNet v2, NasNet, Xception and DenseNet). The only change would be during the second step where we apply transfer learning on the dataset.

5.4 Algorithms and Architectures

5.4.1 VggNet

When a completely connected layer is attached to the network's end, it produces an N-dimensional vector, where N is the number of classes from which the model chooses the appropriate one.

A 224*224 RGB image is used as the input to the VGG-based convNet. The preprocessing layer subtracts the mean image values derived for the complete ImageNet training set from the RGB image with pixel values in the range of 0–255. After preprocessing, the input photos are transmitted via these weight layers. A stack of convolution layers is used to send the training images through. In the VGG16 architecture, there are a total of 13 convolutional layers and three fully connected layers. Instead of having huge filters, VGG features smaller filters (3*3) with better depth. It now has the same effective receptive field as if only one 7 × 7 convolutional layers were used.

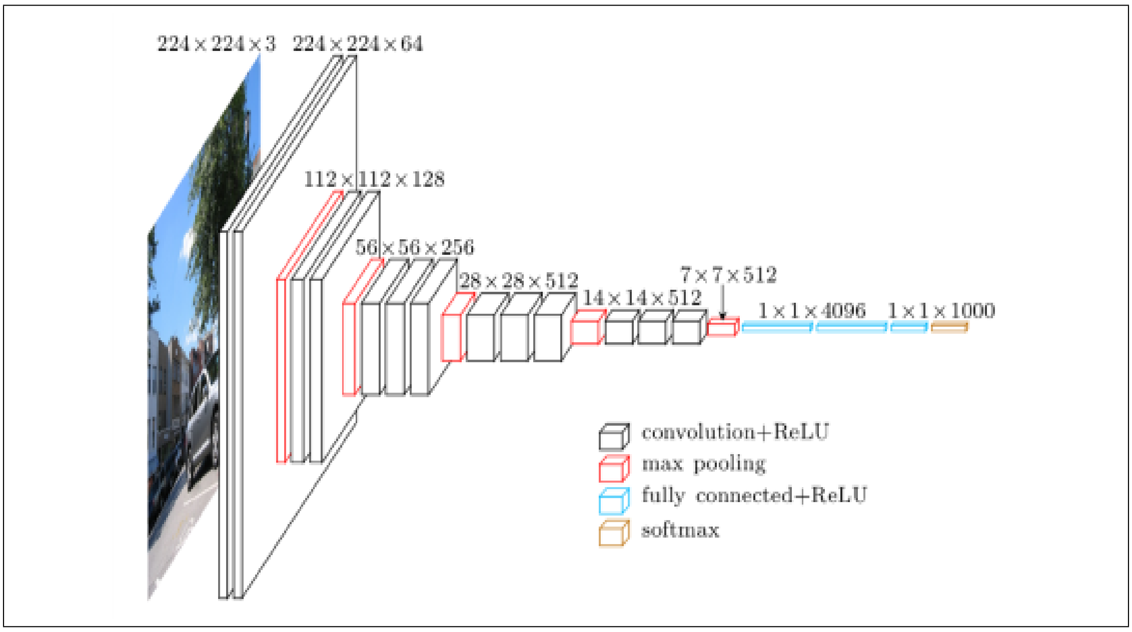


Figure 2: VggNet Architecture

Another VGGNet variant has 19 weight layers, with 16 convolutional layers and 3 fully connected layers, as well as the same 5 pooling layers. VGGNet has two completely connected layers with 4096 channels each, followed by another fully connected layer with 1000 channels to predict 1000 labels in both variations. Softmax layer is used for categorization in the last fully linked layer.

Figure 3: Vgg16 and Vgg19 architecture

5.4.2 Residual Neural Network (ResNet)

A residual neural network (ResNet) is a type of artificial neural network (ANN) that is based on pyramidal cell constructions in the cerebral cortex. Skip connections, or shortcuts, are used by residual neural networks to jump past some layers. The majority of ResNet models use double- or triple-layer skips with nonlinearities (ReLU) and batch normalisation in between. To learn the skip weights, an additional weight matrix can be utilised; these models are known as HighwayNets. DenseNets are models that have several parallel skips. A non-residual network is referred to as a plain network in the setting of residual neural networks.

There are two main reasons to add skip connections: to avoid vanishing gradients, and to mitigate the Degradation (accuracy saturation) problem, in which adding more layers to a sufficiently deep model leads to higher training error.[1] During training, the weights adapt to mute the upstream layer[clarification needed], and amplify the previously-skipped layer. Only the weights for the neighbouring layer's link are changed in the simplest instance, with no explicit weights for the upstream layer. When a single nonlinear layer is stepped over, or when the intermediate layers are all linear, this method works well.

In the early phases of training, skipping layers effectively simplifies the network by employing fewer layers. Because there are fewer layers to propagate through, the influence of disappearing gradients is reduced, which speeds up learning. As the network learns the feature space, it gradually recovers the skipped levels. When all layers are extended at the end of training, it stays closer to the manifold and so learns faster. The feature space is explored further by a neural network with no remnant pieces. This makes it more susceptible to disturbances that lead it to depart off the manifold, and thus takes additional training data to recover.

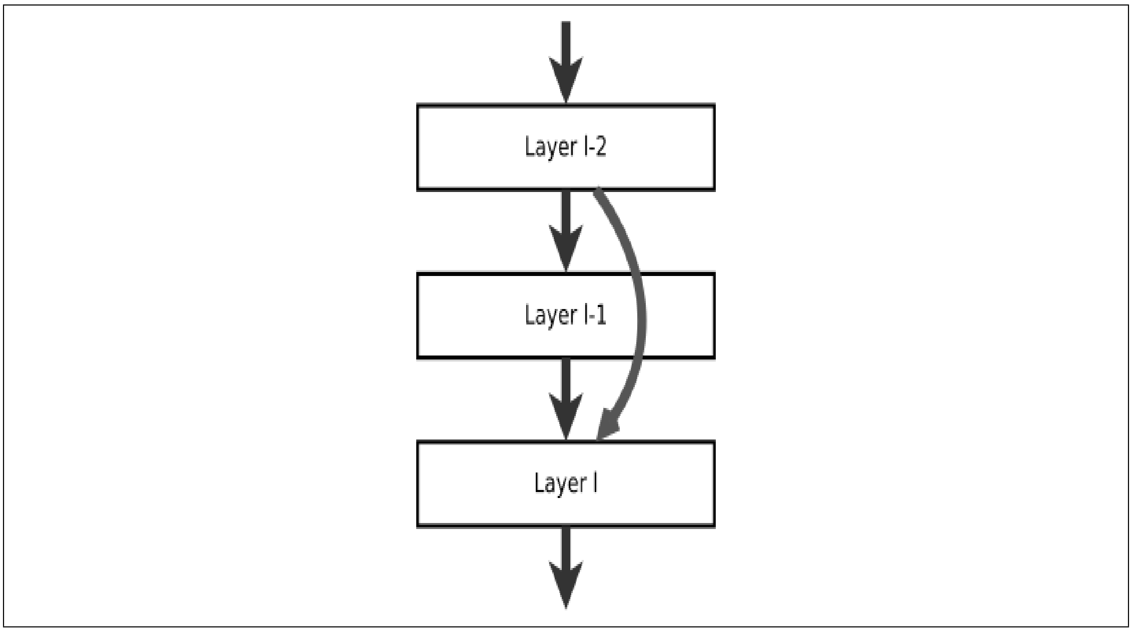


Figure 4: ResNet Connection

5.4.3 NasNet

Google introduced NASNet, which framed the task of finding the ideal CNN architecture as a Reinforcement Learning problem, thanks to its vast computer capacity and engineering talent.

The basic idea was to find the optimal combination of filter sizes, output channels, strides, number of layers, and other characteristics in the specified search space. The accuracy of the searched architecture on the given dataset was the reward for each search action in this Reinforcement Learning environment.

5.4.4 Xception

Francois Chollet proposes the Xception Model. Xception is an expansion of the Inception Architecture that uses depthwise Separable Convolutions to replace the regular Inception modules.

5.4.5 DenseNet

A DenseNet is a sort of convolutional neural network that uses Dense Blocks to connect all layers (with matching feature-map sizes) directly to each other, resulting in dense connections between layers. Each layer takes extra inputs from all preceding levels and passes on its own feature-maps to all following layers to maintain the feed-forward nature.

5.5 Results

The following results were obtained after evaluation of the models of each architecture.

Sr.	Architecture	Speed (in ms)	Accuracy (in %)
1	VggNet16	5699	98.41
2	ResNet	4726	98.12
3	NasNet	5143	93.79
4	Xception	3753	92.34
5	ResNet50	3750	98.27
6	ResNet v2	4031	95.38
7	DenseNet	3892	96.53
8	VggNet19	5421	97.69

Table 2: Results

The graphical representation for the above table:

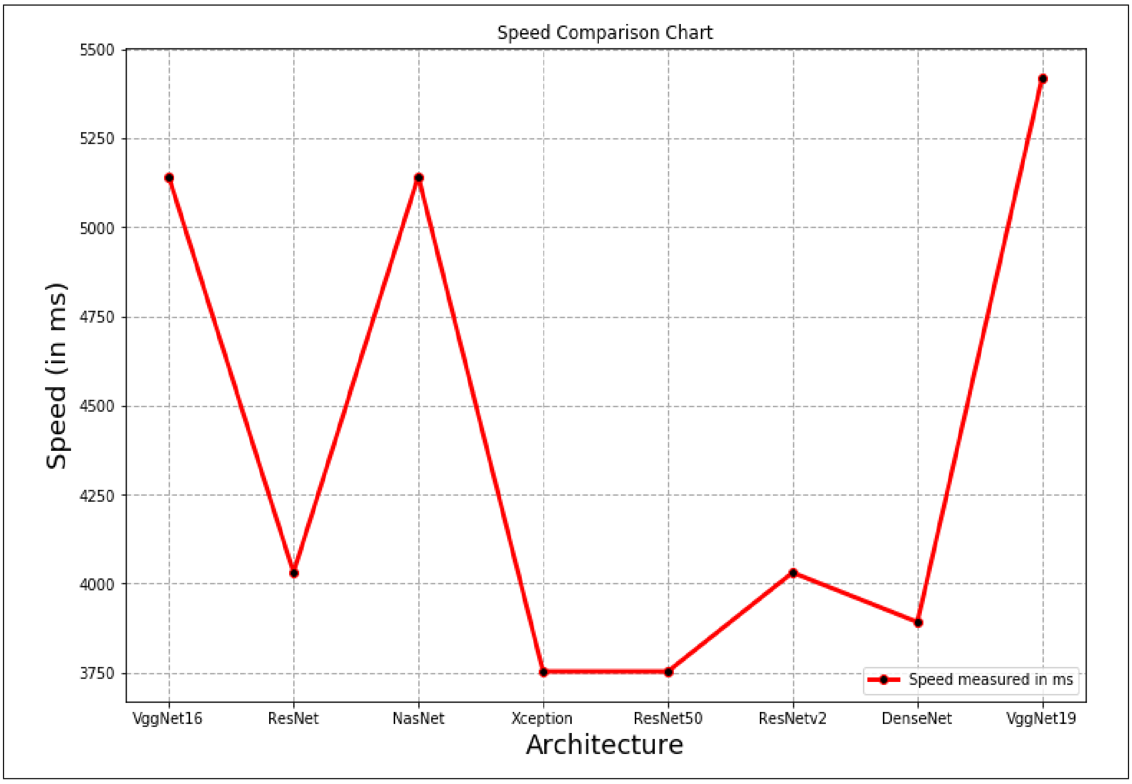


Figure 5: Speed Comparison Chart

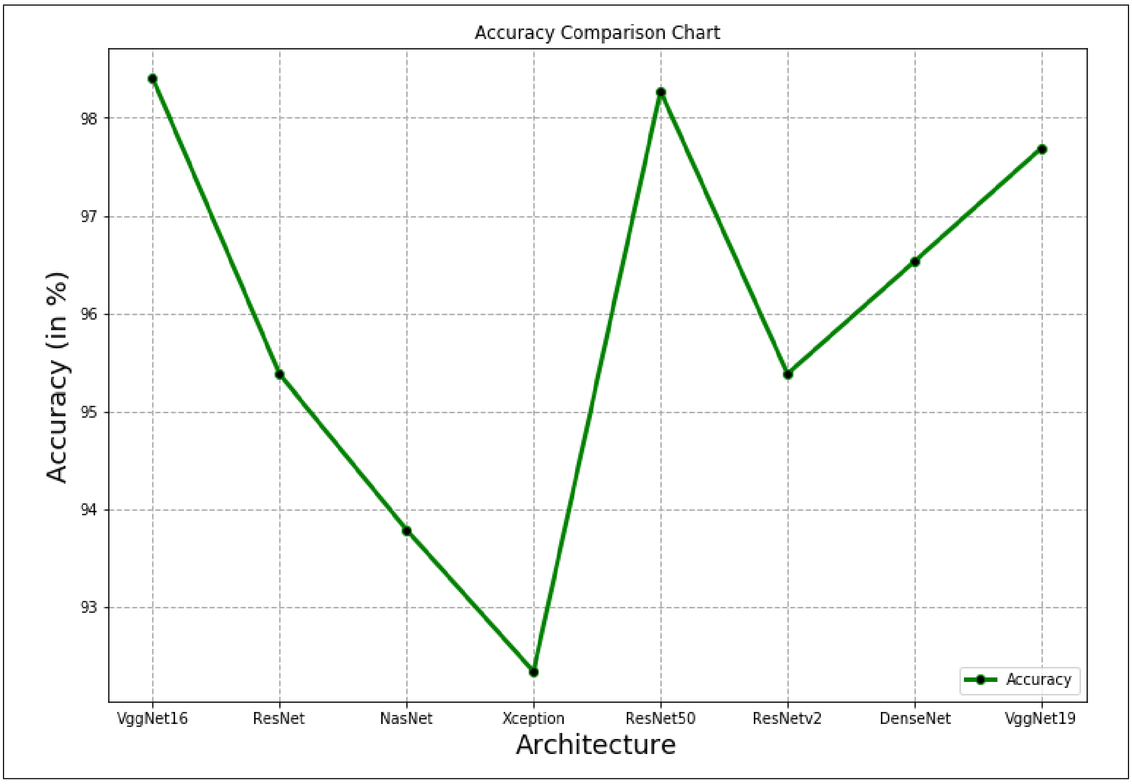


Figure 6: Accuracy Comparison Chart

The fastest algorithm for this problem statement can be identified as ResNet50 and the most accurate can be identified as Vgg16.

6 APPLICATIONS

In the agri-food industry, the proposed system can be used in a variety of ways. This technology can be used by the food processing industry to save human labor, enhance processing speed, and hence increase overall efficiency. In the case of medium-sized food delivery companies, this technology can be used to automate the segregation process and supply the highest quality product to their clients. Farmers can use this technology to separate out rotten food and give the highest quality produce to their customers, addressing the root of the problem. Overall, this approach has the potential to boost agri-food sectors' income and efficiency.

7 CONCLUSION

The goal of this seminar is to learn about the uses of artificial intelligence in the agri-food industry. The transition from human labor to AI-based systems can lower costs, reduce processing time, boost efficiency, and hence boost overall profits. In this seminar, I compared and tested several architectures that could be used to implement the suggested model, determining their speeds and accuracies. The evaluation results revealed the fastest algorithm to be ResNet50 and the most accurate as Vgg16.

References

[1] Yuhang Fu.. "Fruit Freshness Grading Using Deep Learning" *A thesis submitted to the Auckland University of Technology in partial fulfillment of the requirements for the degree of Master of Computer and Information Sciences (MCIS), 2020.*

[2] Nazrul Ismail and Owais A. Malik, "Real-time visual inspection system for grading fruits using computer vision and deep learning techniques", 2020.

[3] Pushpa B R, "Detection and classification of fungal disease in fruits using image processing Techniques", *2015 International Journal of Applied Engineering Research, ISSN 0973-4562 Vol. 10 No.55 (2015).*

[4] BRicha Shah, Pooja Gujarathi, Parth Unadkat, Rijoosinh Mulik and A. N. Bandal, "Quality Analysis of Fruits and Vegetables using Machine Learning Techniques", *International Journal of Research in Engineering, Science and Management (IJRESM) Volume-1, Issue-4, April 2018.*

[5] W. C. Seng and S. H. Mirisaee, "AA New Method for Fruits Recognition System", *2009 International Conference on Electrical Engineering and Informatics, 5-7 August 2009, Selengore, Malaysia.*

[6] Shiv Ram Dubey & Anand Siggh Jalal, A. S., "Detection and Classification of Apple Fruit Diseases using Complete Local Binary Patterns", *3rd IEEE International Conference on Computer and Communication Technology (ICCCT-2012), Volume: 3.*