LOCAL FRUITS RECOGNITION USING CONVOLUTIONAL NEURAL NETWORK

BY

M S Joha Sagor
ID: 163-15-8462

AND

Muhiuddin Dewan
ID: 163-15-8475

This Report Presented in Partial Fulfillment of the Requirements for the Degree of Bachelor of Science in Computer Science and Engineering

Supervised By

Md. Tarek Habib
Assistant Professor
Department of CSE
Daffodil International University

DAFFODIL INTERNATIONAL UNIVERSITY

DHAKA, BANGLADESH

SEPTEMBER 2020

APPROVAL

This Project/internship titled **"Local Fruits Recognition Using Convolutional Neural Network"**, submitted by Muhiuddin Dewan, ID No: 163-15-8475 and M S Joha Sagor, ID No: 163-15-8462 to the Department of Computer Science and Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Engineering and approved as to its style and contents. The presentation has been held on 7 October 2020.

BOARD OF EXAMINERS

Dr. Syed Akhter Hossain **Chairman**
Professor and Head
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Nazmun Nessa Moon **Internal Examiner**
Assistant Professor
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Gazi Zahirul Islam **Internal Examiner**
Assistant Professor
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Dr. Md. Saddam Hossain **External Examiner**
Assistant Professor
Department of Computer Science and Engineering
United International University

DECLARATION

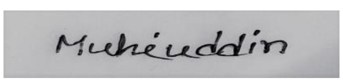
We hereby declare that, this project has been done by us under the supervision of **Md. Tarek Habib, Assistant Professor, Department of CSE** Daffodil International University. We also declare that neither this project nor any part of this project has been submitted elsewhere for award of any degree or diploma.

Supervised by:

____

Md. Tarek Habib
Assistant Professor
Department of CSE
Daffodil International University

Submitted by:



____

(Muhiuddin Dewan)
ID: -163-15-8475
Department of CSE
Daffodil International University

____

(M S Joha Sagor)
ID: -163-15-8462
Department of CSE
Daffodil International University

ACKNOWLEDGEMENT

First, we express our heartiest thanks and gratefulness to almighty God for His divine blessing makes us possible to complete the final year project.

We really grateful and wish our profound our indebtedness to **Md. Tarek Habib**, Assistant Professor, Department of CSE Daffodil International University, Dhaka. Deep Knowledge & keen interest of our supervisor in the field of Computer Vision to carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts and correcting them at all stage have made it possible to complete this project.

We would like to express our heartiest gratitude to **Dr. Syed Akhter Hossain**, Professor and Head**,** Department of CSE, for his kind help to finish our project and also to other faculty member and the staff of CSE department of Daffodil International University.

We would like to thank our entire course mate in Daffodil International University, who took part in this discuss while completing the course work.

Finally, we must acknowledge with due respect the constant support and patience of our parents.

ABSTRACT

Recent advancement of computer vision has made object detection from images much easier, however automatically classifying fruits using computer vision still remains a challenging task due to similarities between different types and various factors like their position (e-g stacked) or lighting conditions. A fruit classification system can play a vital role in major fields like autonomous agricultural robotics or simply be used in developing mobile applications for detecting specific fruit species on the market. In this paper, we evaluated 5 different models that used deep convolutional neural network (DCCN) techniques for fruit detection and proposed an efficient model based on our training results. VGG-16, RESNETV2-152, INCEPTION-V3, XCEPTION, DENSENET-201 was used to train with fruits that are endemic to Bangladesh. Our dataset contained fruit images belonging form 7 different class of native fruits. 80 percent of the dataset was utilized for training and the other 20 percent for testing purposes. The training dataset was augmented and increased for better training convenience. We conducted the experiment with our own dataset and the VGG-16 model achieved a high accuracy rate of 100%.

Table of Contents

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

1.1 Introduction

Gradual advancement over the years in Convolutional Networks has made major contributions in the area of image recognition. Utilizing Deep Neural Networks for identification, classification and differentiate between different classes of fruits outperform all other techniques and algorithms. However, fruit classification can be difficult for numerous reasons like shape, colour and texture similarity among fruits belonging to the same class. [17] In this paper we evaluated five deep learning models with our dataset and determined which gave satisfactory results.

The Inception architecture [7] which was later refined as Inception V3 [8] has been one of the best performing models on both the ImageNet dataset [5] and JFT dataset [6]. The Xception architecture [12] inspired by the Inception architecture, replaced the Inception modules with depth wise separable convolutions outperforming Inception V3 on a smaller margin with the ImageNet dataset and on a significantly larger margin with the greater image classification dataset.

The VGG-16 model [9] is a 16 layers deep convolutional neural network which accomplishes a 92.7% top 5 test accuracy in ImageNet dataset. It focuses on the increased depth of the layers for significant results. However, deeper neural networks are difficult to train because of the gradient descent issue. The Resnet model [13] tackled this problem by utilizing shortcut connections. This model achieves 3.57% error on the ImageNet test set. The DenseNet model is a novel architecture by Huang et al. [14] which focuses more on shortcut connections by connecting the layers in a feed forward fashion.

1.2 Motivation

Automatically detecting fruits using computer vision still remains a challenging task due to similarities between colours and textures as well as different types and factors like, their position or lighting conditions. We want to evaluate 5 well known architectures that use deep convolutional neural network for image detection and use them for detecting our Bangladeshi local fruits. With this approach we will be able to test the efficiency of these models as well as their performance on our own Bangladeshi fruits, both of which are novel approaches.

1.3 Research Questions

- How Fruit Recognition Impacts on Our Everyday Life?
- Can we simultaneously apply machine learning and deep learning models?
- What are the challenges faced by Fruit Recognition applications?
- Which approach is better for Fruit Recognition?
- What are the areas of research in Fruit Recognition, where I can explore my research topics?

1.4 Expected Outcome

Fruit recognition is a kind of fine-grained visual recognition which is a relatively harder problem than conventional image recognition. Classification of fruits automatically, has both academic and industrial potential. If trained with a diverse database it could be used in smartphone applications that detect species of fruit instantly and could provide information like its price, nutritional benefits, calories etc. Moreover, it would be able to detect species of fruits that are endemic to a specific region.

CHAPTER 2
BACKGROUND

2.1 Introduction:

Convolutional networks don’t perceive images like humans rather they perceive them as volumes. We see an image in two dimensions, width and height whereas convolutional networks also perceive depth. Because of the RGB encoding of an image the convolutional network ingests an image as three separate depth layers stacked one top of another, which are referred to as channels.

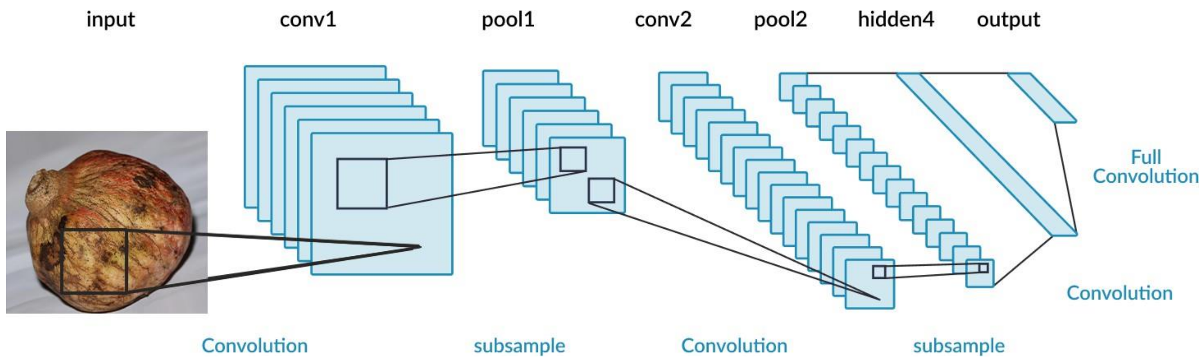


Figure 2.1: CNN Architecture.

The convolutional network takes square patches of pixels and passes them through a filter. A filter is also known as kernel and it finds patterns in the pixels. An activation map is equivalent to the number of steps the filter is required for traversing an underlying image. Images contain pixels with various patterns. These patterns create additional activation maps resulting a new volume. Since the images have higher dimensions, they require high computing power and time in order to process. Convolutional networks solved this problem by reducing the dimensionality in ways like filter stride and down sampling.

2.1.1 CNN Training Algorithm:

There are two stages involved in the training.

Forward Propagation:

A sample (X, Op) is picked from the sample set, here X refers to the input of the network. The network calculates the actual output Ip. The neural network learns new feature space by computing linear transformations of the given inputs first and after that applying non-linear functions to them. The inputs are then transferred into the next layer. The information travels from input layers through hidden layers to output layers in this stage.

Backward Propagation:

The actual output Op and the ideal output Ip is calculated and the weight matrix of the neural net is adjusted based on the error rate obtained from previous epoch. This stage helps calculating the gradient loss of a function.

2.2 Related Work

The environmental factors like lighting, shadows and reflections as well as factors like physical appearance of the fruit such as colour, texture and its size make the process of fruit recognition very challenging therefore many researchers view the task as an image segmentation problem. Wang et al. 2012 [11] worked on apple detection based on their colour and reflection pattern for yield estimation. Yamamoto et al. [10] used colour-based segmentation for detecting tomatoes.

Several works have been done based on deep neural networks for tackling the fruit recognition problem. Lei Hou 2016 [1] proposed an algorithm that uses a combination of CNN algorithm and selective search algorithm. Their database contained a small number of 5330 images with 7 different classes with a recognition rate of 99.77%. They used 4000 images for training and 1330 for testing. Inkyu Sa 2016 [2] proposed a novel approach for detecting fruits from images utilizing deep neural networks. Their objective was to create a neural network that would benefit autonomous robots capable for harvesting fruits. They used a Region-based CN for this purpose. Their training sample consisted of RGB and NIR (near infra-red) images and they combined these to 2 separate cases which resulted in a multi modal network that gave significantly better results than the existing networks. Ishrar Hussain 2018 [3] proposed another novel approach based on DCCN, considering the external environmental changes and real-world changes. Their two-track deep neural model architecture obtained a high accuracy of 99% on their own dataset containing 44406 images belonging from 15 different categories of fruits. Another work of Shamim Hossain 2018 [4] demonstrates automatic fruit classification using deep learning for industrial applications. Their proposed framework is established on two different deep learning architectures, one with six convolutional neural network layers and the other with a visual geometry group-16 pre-trained model respectively. Their experiments on two colour-image datasets resulted in a classification accuracy of 99.49% and 99.75% on the first dataset and 85.43% and 96.75% accuracy on the second one. Keiji Yanai 2015, used DCNN for food photo recognition, which is similar to fruit detection in several ways. [15] They used techniques such as, pre-training, fine-tuning and activation features from pre-trained DCNN. They achieved 78.77% accuracy on UEC_FOOD100 by pre-training the fine-tuned DCNN with 2000 categories with ImageNet dataset [5]. Srikanth Tammina 2019 [16] proposed transfer learning or reusing a pretrained model like VGG-16 for classifying images. VGG-16 model [9] with fine-tuned CNN and image augmentation achieved an accuracy of 95.40%.

CHAPTER 3
RESEARCH METHODOLOGY

3.1 Research Methodology:

Research Methodology is the main part of the work of a research project. Because by the methodology, a research project is able to move forward step by step and can reach the goal with perfection. By the correct methodology and proper planning, it is also possible to discover new things. We have also worked on fruit recognition dataset through various procedure. The procedure consists of Data collection, Data preprocessing, Data Augmentation, Train Model (VGG-16, RESNETV2-152, INCEPTION-V3, XCEPTION, DENSENET-201), Model Evaluation.

Work Flow for Fruit Recognition System:

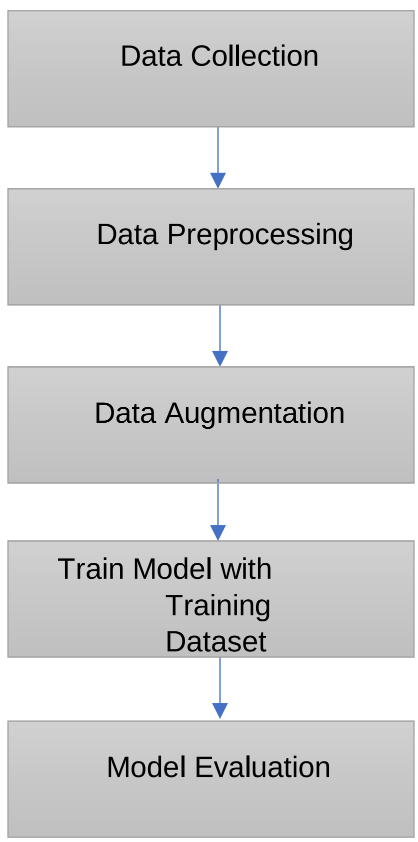


Figure 3.1: Work Flow Diagram.

3.1.1 Data Collection**:**

In this project we have recognize Bangladeshi local fruits from images. We have used multiple local fruits images for this purpose. Since there is no dataset available for Bangladeshi local fruits image, so we need to create a new data set for this project. We have checked many supermarkets with personals and collected 7 categories local fruits image (amloki, ata, bangi, bel, chalta, orboroi, sofeda) from super market. But it is a matter of sorrow that we were unable to gather enough local fruits images. Because of the pandemic situation of the country and the fruits were also unavailable on that time.

3.1.2 Data Preprocessing:

Data is dissipated in a dataset, making it hard to get exact outcomes. Data preprocessing is important for that reason. Different boundaries through which the real outcomes can be acquired when the expectation is performed have been balanced from this dataset which for it can improve to get real precision and achieve better execution.

3.1.3 Data Augmentation:

Data expansion is a system that engages experts to basically fabricate the various assortment of information open for getting ready models, without truly assembling new information. We have utilized data enlargement procedures, for example, zoom, shear, flip, move, splendor to prepare our convolutional neural network models.

3.1.4 Train Model with training dataset:

We have selected five different convolutional neural network model and train them with our training dataset which is contained 4227 local fruits image. The models we choose are:

- DENSENET-201
- XCEPTION
- RESNETV2-152
- VGG-16
- INCEPTION-V3

3.1.5 Model evaluation:

Model evaluation is a core apart to make an effective convolutional neural network model. We have used Confusion matrix, Accuracy, F1-score, Precision, Recall and plot diagram for evaluating our model.

3.2 VGG-16 Model:

The VGG-16 model comprises of 16 weight layers. These layers contain 13 convolutional layers with a filter size of 3x3 and three fully connected layers. These convolutional layers are partitioned into five groups where each group is trailed by a max pooling layer. The filters of the convolutional layer group begin with 64 and increases by a factor of 2, until it arrives at 512. The input layer of the VGG-16 model takes a 224 x 224 fixed size RGB image and after that the image goes through a pile of convolutional layers. It can also be configured to utilize 1x1 convolutional filter. Here the convolutional stride is fixed at 1 pixel where the spatial padding of this layer's input is such that the spatial resolution is preserved after convolution. The five max-pooling layers do the spatial pooling, which follow a portion of the convolutional layers and with stride 2 and over a 2x2 pixel window max pooling is performed.

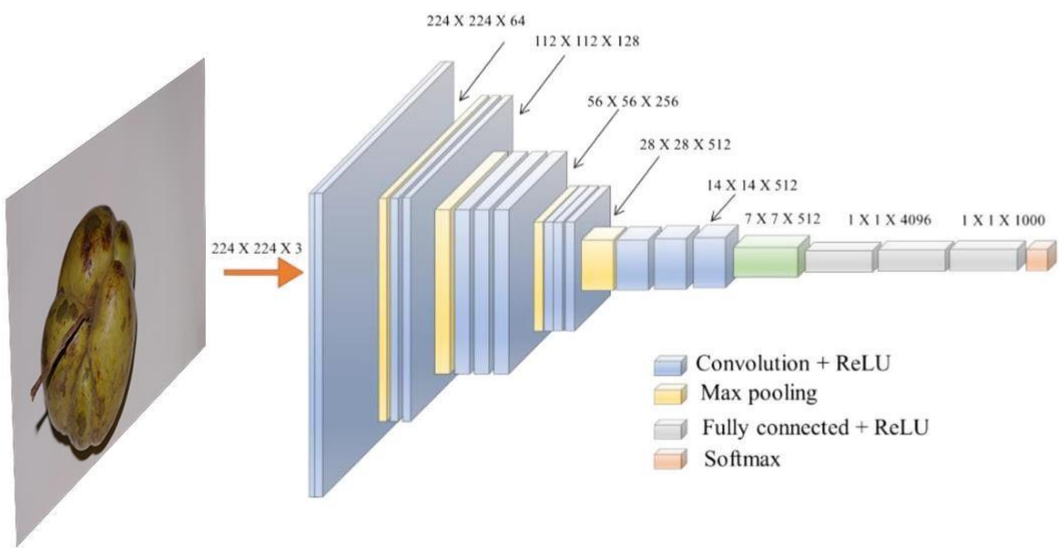


Figure 3.2: VGG-16 architecture diagram.

Following the stack of convolutional layers there are three fully connected layers where the first two of them consist of 4096 channels each and the third layer contains seven channels, one for each class. The final layer is known as the soft max layer. The fully connected layers in all the networks are configured similarly.

3.3 RESNET Model:

Residual networks or RESNET has a similar architecture to the VGG-16 however, resnet provided a better solution to the notorious vanishing gradient problem by its identity mapping capability. It won the first place in the ILSVRC and COCO 2015 in the ImageNet and Coco challenge. Resnet introduced the identity shortcut connection which enabled us to train deep neural networks with layers up to 150 and more successfully. Beside solving the vanishing gradient problem, the shortcut connection enables the model to learn an identity function. This function ensures the performance of the upper layers to be on par with the lower ones.

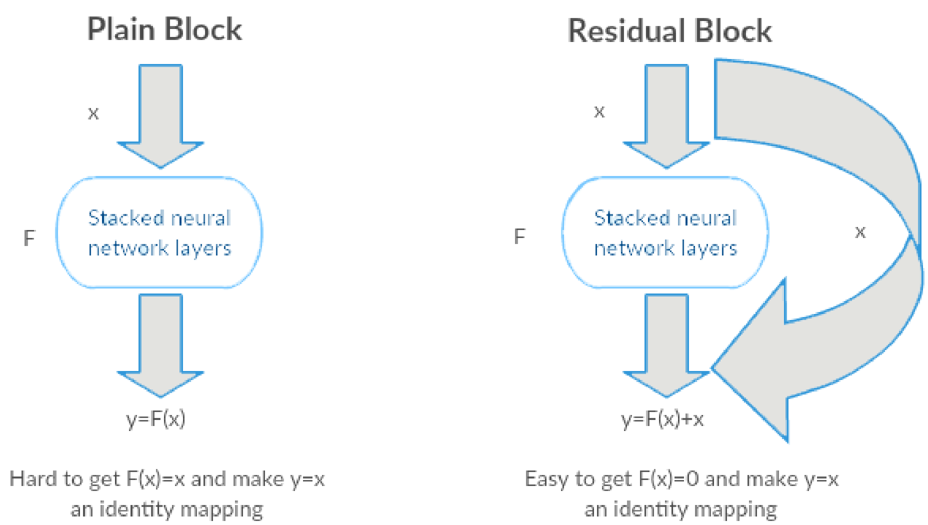


Figure 3.3: RESNET Identity Mapping.

3.4 DenseNet Model:

As network layers get deeper, the gradients aren't adequately back propagated to the initial layers of the network, this is called the gradient descent problem. For this problem the initial layers are unable to learn the low-level features from the underlying layers as the gradients keep getting smaller as they move backwards. All the CNN architectures tackle this issue by creating a path for the information to transmit between the initial layers and the final layers, DenseNet architecture solves this problem by creating channels between the layers of the network.

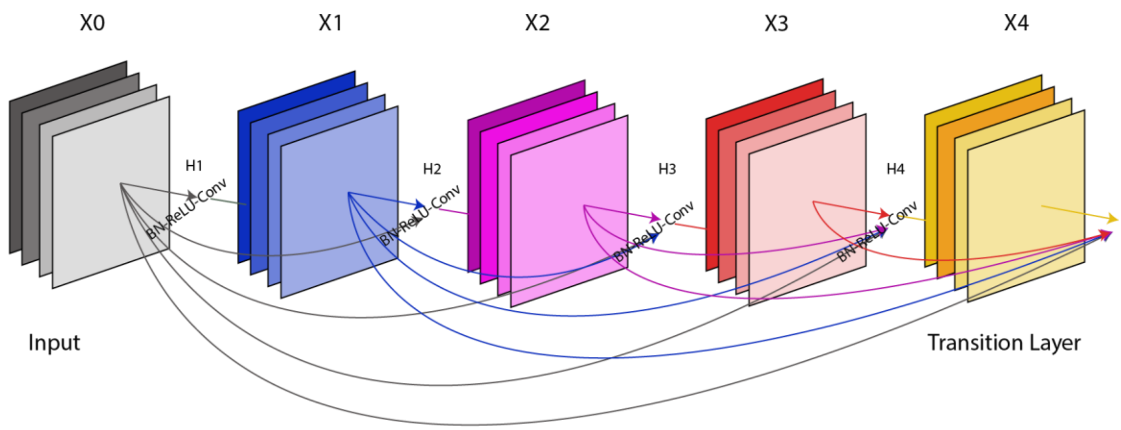


Figure 3.4: Dense Net Architecture Diagram.

In the DenseNet architecture, the information flows between the layers, each layer in a dense block receives feature maps from all the previous layers and passes its output to the later layers. A dense block contains n dense layers and these layers are interconnected. The dimensions of the features remain constant in a dense block. Each of these dense-layers contain a 1x1 conv and a 3x3 conv operation where 1x1 is used for feature extraction and 3x3 is for feature depth count respectively, moreover each of these layers can directly access loss function gradients along with the original input signal. Unlike RESNET architecture the feature maps received from the subsequent layers are merged through concatenation, not summation. These dense connections enable the architecture to require fewer layers and reuse the gathered knowledge. There are many variations of the DenseNet model with fewer and narrower layers which results to reduced parameter count therefore it is much easier to train them.

3.5 Inception V3:

When compared to VGG architecture, Inception architecture can be considered more computationally efficient in terms of number parameters and resource management. Inception v3 modifies the earlier inception architectures in order to achieve efficiency by using less computational power. Changes made in the Inception network for different use cases can be challenging due to the risk of losing computational efficiency. Techniques such as factorized convolutions, dimension reduction, regularization and parallel computations are utilized in the architecture for easier model adaptation.

Factorizing Convolutions:

This technique alleviates the number of parameters involved in a network without decreasing the network efficiency. In case of smaller convolutions, replacing the bigger convolutions with smaller ones can achieve less training time. For example: Two 3x3 convolutions of 18 parameters replacing a 5x5 convolution with 25 parameters, which means less parameters for the model to deal with. When considering Asymmetric Convolutions, a 3x3 convolution is replaced by one 3x1 convolution followed by one 1x3 convolution. The 3x3 convolution generates 9 parameters which is replaced by 6 (3x1+1x3) parameters.

Auxiliary classifier:

An auxiliary classifier can be described as a small convolutional neural network which is inserted between layers during training. They were used for achieving deeper network in the Inception V1 however, in Inception V3 only one auxiliary classifier is used and it acts as a regularizer.

Reducing grid size:

The feature map downsizing/ grid size reduction is usually done by max pooling. Due to drawbacks such as too greedy or too expensive, more efficient grid size reduction can be used.

3.6 Xception Model:

Xception model outperforms the Inception model by utilizing a modified depth wise separable convolution. Xception model can be considered the more extreme interpretation of the Inception architecture. The depth wise convolution can be defined as channel wise nxn spatial convolution e.g. If we have 3 channels, then we will have 3 nxn spatial convolution. Pointwise convolution is the 1x1 convolution used for changing dimension.

The Xception model uses the modified depth wise separable convolution which is the pointwise convolution followed by a depth wise convolution whereas originally depth wise separable convolution was actually the opposite of that. This modification performs the 1x1 convolution first and then nxn spatial convolutions take place. For this change performing convolutions across all channels is no longer necessary therefore, the model becomes much lighter with fewer connections. The overall architecture of the Xception model can be divided into 3 sections, Entry flow, Middle flow and Exit flow. Modified depth wise separable convolutions are treated as inception modules in the architecture and appear in all three sections along with the residual connections.

CHAPTER 4
EXPERIMENTAL RESULTS AND DISCUSSION

4.1 Introduction:

This chapter demonstrates our training and testing results using the five DCNN models mentioned above. We separated the tests to five different tests for each model and below are the results achieved from each of them. We used a total number of 4345 fruit images from which 4227 were reserved for training the models and 118 was reserved for testing. We enlarged our training and testing data by augmentation for better results.

Sample: A sample can be considered a single row of data.

Batch: A batch size refers to the number of samples to work with before updating the parameters of the internal models.

Epoch: Epochs are hyper-parameters that measure the number of times the training model will work on the entire training dataset.

Confusion Matrix: A confusion Matrix is a performance measurement for machine learning classification models on a test dataset.

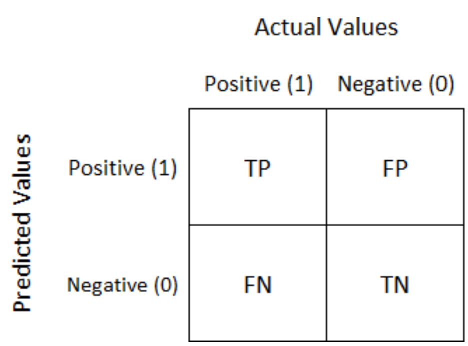


Figure 4.1: Confusion Matrix.

Data Samples

Fruit Name	Total Images
Amloki	468
Ata	773
Bangi	758
Bel	590
Chalta	548
Orbori	607
Sofeda	601

Figure 4.2: Sample Training Images.

4.2 Experimental Result and Analysis for DenseNet-201 Model:

Parameter:

Total parameters: 18,980,551.

Trainable parameters: 658,567.

Non-trainable parameters: 18,321,984.

Accuracy:

Accuracy stated for each number of epochs. Here, Increasing the number of epochs results in accuracy reduction.

```
Epoch 1/10
133/133 [==============================] - 2492s 19s/step - loss: 0.5544 - accuracy: 0.9375 - val_loss: 1.0183 - val_accuracy: 0.9153
Epoch 2/10
133/133 [==============================] - 2993s 23s/step - loss: 0.1455 - accuracy: 0.9851 - val_loss: 1.4511 - val_accuracy: 0.9407
Epoch 3/10
133/133 [==============================] - 2992s 22s/step - loss: 0.0518 - accuracy: 0.9931 - val_loss: 1.7852 - val_accuracy: 0.9068
Epoch 4/10
133/133 [==============================] - 2989s 22s/step - loss: 0.1359 - accuracy: 0.9882 - val_loss: 3.5451 - val_accuracy: 0.8983
Epoch 5/10
133/133 [==============================] - 2987s 22s/step - loss: 0.1246 - accuracy: 0.9896 - val_loss: 4.2295 - val_accuracy: 0.7797
Epoch 6/10
133/133 [==============================] - 2987s 22s/step - loss: 0.0341 - accuracy: 0.9960 - val_loss: 5.8369 - val_accuracy: 0.8475
Epoch 7/10
133/133 [==============================] - 2985s 22s/step - loss: 0.0270 - accuracy: 0.9969 - val_loss: 2.6701 - val_accuracy: 0.8559
Epoch 8/10
133/133 [==============================] - 2985s 22s/step - loss: 0.2215 - accuracy: 0.9858 - val_loss: 5.7421 - val_accuracy: 0.8220
Epoch 9/10
133/133 [==============================] - 2989s 22s/step - loss: 0.0738 - accuracy: 0.9946 - val_loss: 10.3661 - val_accuracy: 0.7203
Epoch 10/10
133/133 [==============================] - 2986s 22s/step - loss: 0.0462 - accuracy: 0.9957 - val_loss: 9.2603 - val_accuracy: 0.7966
```

Figure 4.3: Accuracy Diagram for DenseNet-201.

Confusion Matrix:

States the accuracy of identification. Here, Accurate identification of Amloki and Orboroi fruit.

```
Confusion Matrix
[[ 8  0  0  0  0  0  0]
 [ 0  7  0 11  0  0  0]
 [ 0  0 19  1  0  0  0]
 [ 0  0  0 23  0  0  3]
 [ 0  0  0  6 10  0  0]
 [ 0  0  0  0  0 14  0]
 [ 2  0  0  1  0  0 13]]
```

Figure 4.4: Confusion Matrix for DenseNet-201.

Classification Report:

Shows precision, recall, f1-score and accuracy between the fruit classes. Achieved an accuracy rate of 80% from this model.

```
Classification Report
              precision    recall  f1-score   support

      amloki       0.80      1.00      0.89         8
         ata       1.00      0.39      0.56        18
       bangi       1.00      0.95      0.97        20
         bel       0.55      0.88      0.68        26
      chalta       1.00      0.62      0.77        16
      orbori       1.00      1.00      1.00        14
      sofeda       0.81      0.81      0.81        16

    accuracy                           0.80       118
   macro avg       0.88      0.81      0.81       118
weighted avg       0.86      0.80      0.79       118
```

Figure 4.5: Classification Report for DenseNet-201.

Train Accuracy Vs Validation Accuracy:

We notice a lower difference in the beginning and gradually larger difference in the progression.

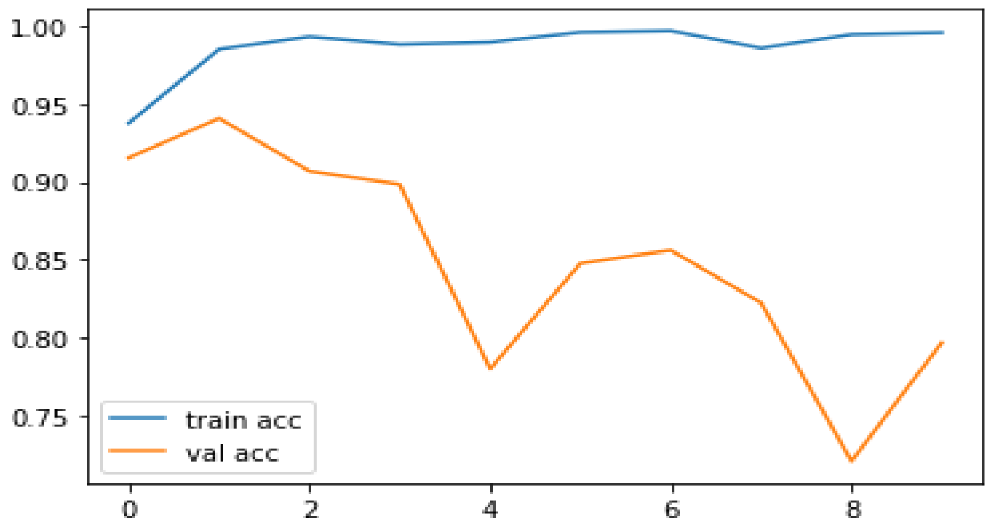


Figure 4.6: Train Accuracy vs Validation Accuracy for DenseNet-201.

Train Loss Vs Validation Loss:

There is no noticeable train loss however validation loss curve fluctuated immensely.

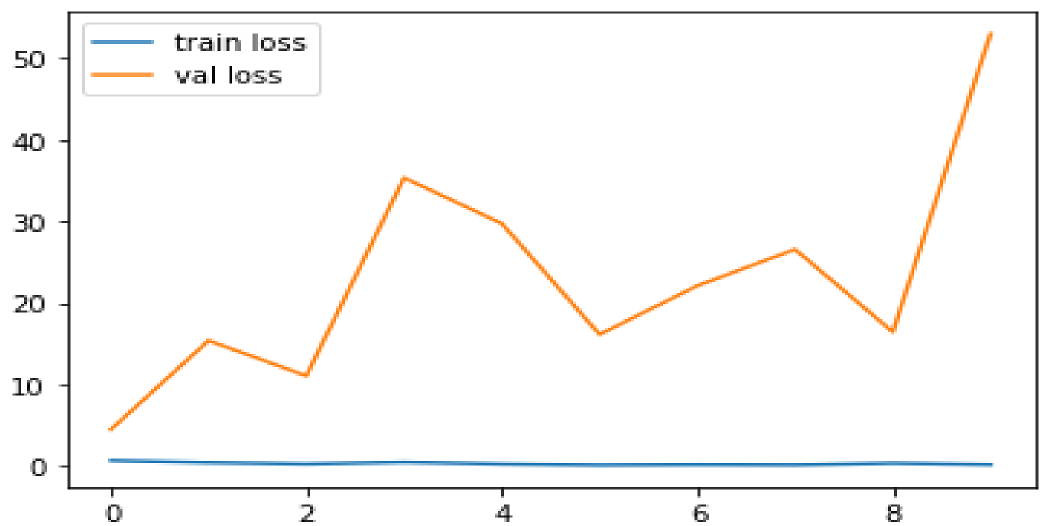


Figure 4.7: Train Loss vs Validation Loss for DenseNet-201.

4.3 Experimental Result and Analysis for InceptionV3 Model:

Parameter:

Total parameters: 22,161,191.

Trainable parameters: 358,407.

Non-trainable parameters: 21,802,784.

Accuracy:

Accuracy stated for each number of epochs. Here, Increasing the number of epochs results in increased accuracy.

```
Found 4227 images belonging to 7 classes.
Found 118 images belonging to 7 classes.
Epoch 1/10
133/133 [==============================] - 1290s 10s/step - loss: 0.9689 - accuracy: 0.8680 - val_loss: 1.7902 - val_accuracy: 0.6441
Epoch 2/10
133/133 [==============================] - 1242s 9s/step - loss: 0.2654 - accuracy: 0.9567 - val_loss: 8.3659 - val_accuracy: 0.6102
Epoch 3/10
133/133 [==============================] - 1232s 9s/step - loss: 0.1433 - accuracy: 0.9754 - val_loss: 11.6956 - val_accuracy: 0.7542
Epoch 4/10
133/133 [==============================] - 1227s 9s/step - loss: 0.1854 - accuracy: 0.9714 - val_loss: 24.2778 - val_accuracy: 0.8305
Epoch 5/10
133/133 [==============================] - 1228s 9s/step - loss: 0.2142 - accuracy: 0.9747 - val_loss: 10.7668 - val_accuracy: 0.8051
Epoch 6/10
133/133 [==============================] - 1232s 9s/step - loss: 0.2130 - accuracy: 0.9785 - val_loss: 11.6239 - val_accuracy: 0.7712
Epoch 7/10
133/133 [==============================] - 1225s 9s/step - loss: 0.3129 - accuracy: 0.9721 - val_loss: 21.1950 - val_accuracy: 0.7627
Epoch 8/10
133/133 [==============================] - 1227s 9s/step - loss: 0.2820 - accuracy: 0.9780 - val_loss: 14.4957 - val_accuracy: 0.8898
Epoch 9/10
133/133 [==============================] - 1227s 9s/step - loss: 0.2395 - accuracy: 0.9813 - val_loss: 23.0632 - val_accuracy: 0.7881
Epoch 10/10
133/133 [==============================] - 1231s 9s/step - loss: 0.2977 - accuracy: 0.9763 - val_loss: 25.3987 - val_accuracy: 0.8644
```

Figure 4.8: Accuracy Diagram for Inception V3.

Confusion Matrix:

States the accuracy of identification. Here, Accurate identification of Amloki, Ata, Bangi, Bel, Chalta, Orbori fruits.

```
Confusion Matrix
[[ 8  0  0  0  0  0  0]
 [ 0 17  0  0  1  0  0]
 [ 0  0 19  1  0  0  0]
 [ 0  0  0 26  0  0  0]
 [ 0  0  0  0 16  0  0]
 [ 0  0  0  0  0 14  0]
 [ 0  0  0  4  0 10  2]]
```

Figure 4.9: Confusion Matrix for Inception V3.

Classification Report:

Shows precision, recall, f1-scrore and accuracy between the fruit classes. Achieved an accuracy rate of 86% from this model.

Classification Report

	precision	recall	f1-score	support
amloki	1.00	1.00	1.00	8
ata	1.00	0.94	0.97	18
bangi	1.00	0.95	0.97	20
bel	0.84	1.00	0.91	26
chalta	0.94	1.00	0.97	16
orbori	0.58	1.00	0.74	14
sofeda	1.00	0.12	0.22	16
accuracy			0.86	118
macro avg	0.91	0.86	0.83	118
weighted avg	0.91	0.86	0.83	118

Figure 4.10: Classification Report for Inception V3.

Train Accuracy Vs Validation Accuracy:

Higher difference in accuracy from the beginning and gradual reduction in difference.

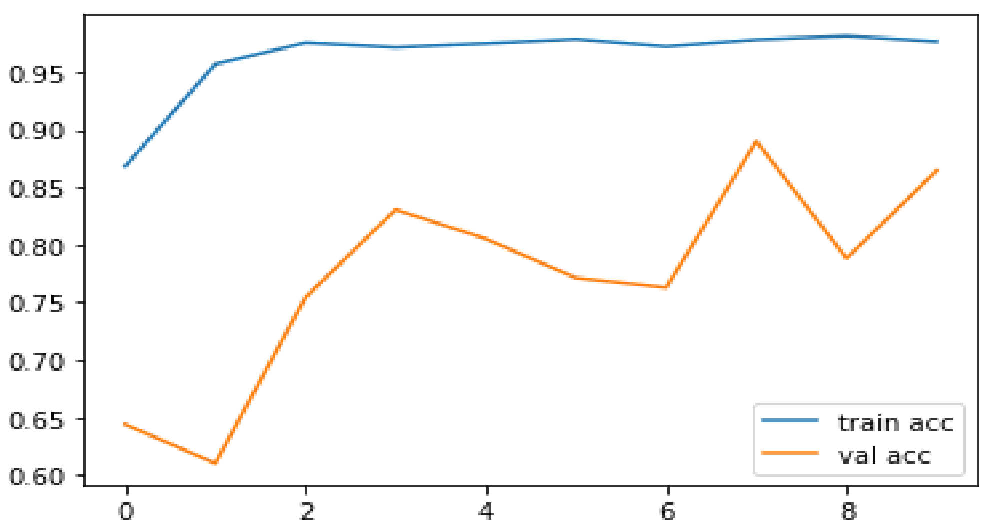


Figure 4.11: Train Accuracy vs Validation Accuracy for Inception V3.

Train Loss Vs Validation Loss:

Validation loss reached its peak and ultimately kept increasing.

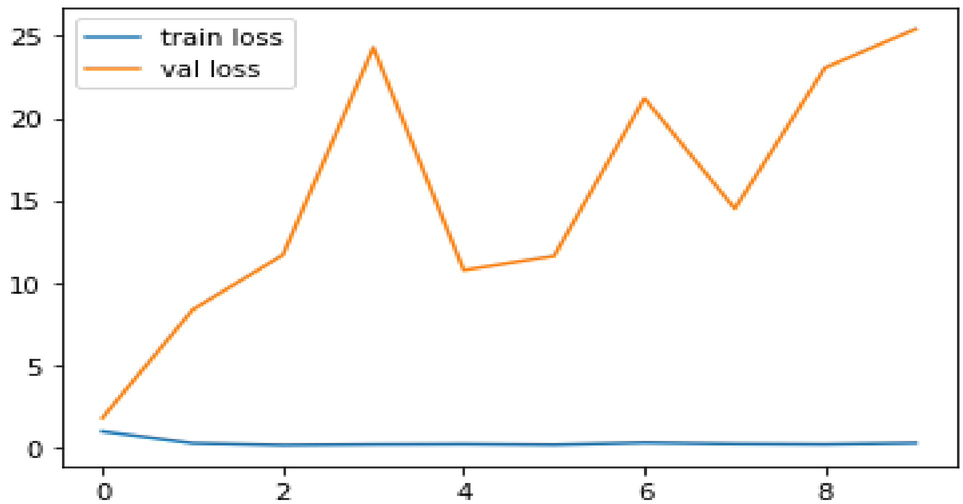


Figure 4.12: Train Loss vs Validation Loss for Inception V3.

4.4 Experimental Result and Analysis for VGG-16 Model:

Parameter:

Total parameters: 14,890,311.

Trainable parameters: 175,623.

Non-trainable parameters: 14,714,688.

Accuracy:

Accuracy stated for each number of epochs. Here, Increasing the number of epochs results in increased accuracy.

```
Found 4227 images belonging to 7 classes.
Found 118 images belonging to 7 classes.
Epoch 1/10
133/133 [==============================] - 1842s 14s/step - loss: 0.3409 - accuracy: 0.8933 - val_loss: 0.1944 - val_accuracy: 0.9915
Epoch 2/10
133/133 [==============================] - 1782s 13s/step - loss: 0.0624 - accuracy: 0.9832 - val_loss: 0.0352 - val_accuracy: 1.0000
Epoch 3/10
133/133 [==============================] - 1787s 13s/step - loss: 0.0393 - accuracy: 0.9882 - val_loss: 0.1042 - val_accuracy: 1.0000
Epoch 4/10
133/133 [==============================] - 1780s 13s/step - loss: 0.0185 - accuracy: 0.9962 - val_loss: 0.0175 - val_accuracy: 1.0000
Epoch 5/10
133/133 [==============================] - 1785s 13s/step - loss: 0.0143 - accuracy: 0.9979 - val_loss: 0.0292 - val_accuracy: 1.0000
Epoch 6/10
133/133 [==============================] - 1785s 13s/step - loss: 0.0123 - accuracy: 0.9981 - val_loss: 0.0057 - val_accuracy: 1.0000
Epoch 7/10
133/133 [==============================] - 1791s 13s/step - loss: 0.0107 - accuracy: 0.9979 - val_loss: 0.0125 - val_accuracy: 1.0000
Epoch 8/10
133/133 [==============================] - 1780s 13s/step - loss: 0.0216 - accuracy: 0.9946 - val_loss: 0.0061 - val_accuracy: 1.0000
Epoch 9/10
133/133 [==============================] - 1769s 13s/step - loss: 0.0094 - accuracy: 0.9979 - val_loss: 0.0029 - val_accuracy: 1.0000
Epoch 10/10
133/133 [==============================] - 1768s 13s/step - loss: 0.0054 - accuracy: 0.9995 - val_loss: 0.0023 - val_accuracy: 1.0000
```

Figure 4.13: Accuracy Diagram for VGG-16.

Confusion matrix:

States the accuracy of identification. Here, Accurate identification of all the fruits.

```
Confusion Matrix
[[ 8  0  0  0  0  0  0]
 [ 0 18  0  0  0  0  0]
 [ 0  0 20  0  0  0  0]
 [ 0  0  0 26  0  0  0]
 [ 0  0  0  0 16  0  0]
 [ 0  0  0  0  0 14  0]
 [ 0  0  0  0  0  0 16]]
```

Figure 4.14: Confusion Matrix for VGG-16.

Classification Report:

Shows precision, recall, f1-score and accuracy between the fruit classes. Achieved an accuracy rate of 100% from this model.

Classification Report

	precision	recall	f1-score	support
amloki	1.00	1.00	1.00	8
ata	1.00	1.00	1.00	18
bangi	1.00	1.00	1.00	20
bel	1.00	1.00	1.00	26
chalta	1.00	1.00	1.00	16
orbori	1.00	1.00	1.00	14
sofeda	1.00	1.00	1.00	16
accuracy			1.00	118
macro avg	1.00	1.00	1.00	118
weighted avg	1.00	1.00	1.00	118

Figure 4.15: Classification Report for VGG-16.

Train Accuracy Vs Val Accuracy:

Gradual reduction in difference between the curves.

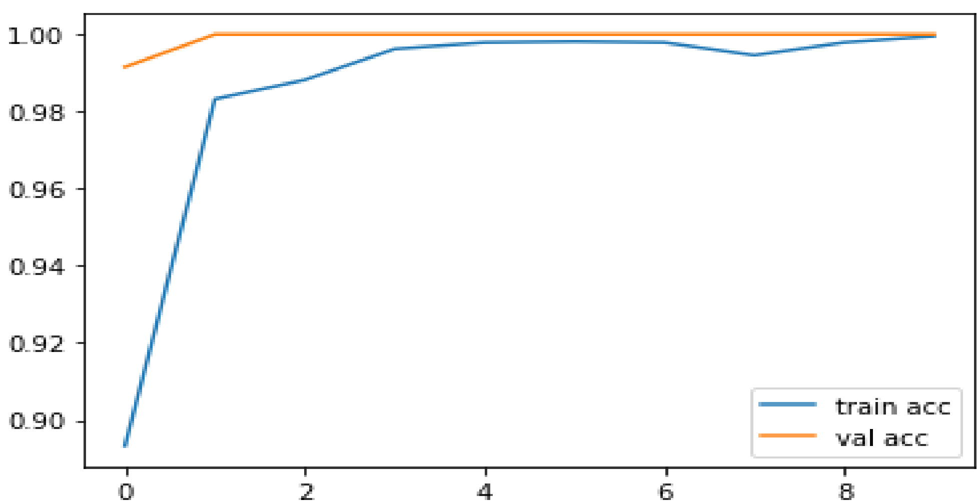


Figure 4.16: Train Accuracy vs Validation Accuracy for VGG-16.

Train Loss Vs Val Loss:

Validation loss fluctuated initially but went lower throughout progression.

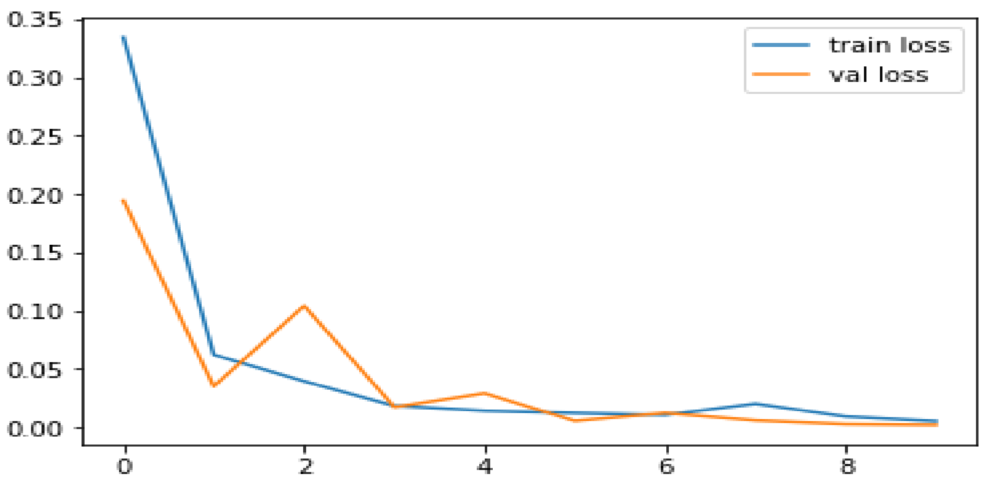


Figure 4.17: Train Loss vs Validation Loss for VGG-16.

4.5 Experimental Result and Analysis for ResNetV2-152 Model:

Parameter:

Total parameters: 59,034,119.

Trainable parameters: 702,471.

Non-trainable parameters: 58,331,648.

Accuracy:

Accuracy stated for each number of epochs. Here, Increasing the number of epochs results in consistent accuracy.

```
Epoch 1/10
133/133 [==============================] - 2604s 20s/step - loss: 0.4901 - accuracy: 0.9456 - val_loss: 10.4027 - val_accuracy: 0.9068
Epoch 2/10
133/133 [==============================] - 2642s 20s/step - loss: 0.2508 - accuracy: 0.9801 - val_loss: 0.0000e+00 - val_accuracy: 0.9322
Epoch 3/10
133/133 [==============================] - 2585s 19s/step - loss: 0.1808 - accuracy: 0.9851 - val_loss: 5.5877 - val_accuracy: 0.9237
Epoch 4/10
133/133 [==============================] - 2560s 19s/step - loss: 0.2017 - accuracy: 0.9884 - val_loss: 3.9863 - val_accuracy: 0.9576
Epoch 5/10
133/133 [==============================] - 2569s 19s/step - loss: 0.1389 - accuracy: 0.9915 - val_loss: 3.1113 - val_accuracy: 0.9492
Epoch 6/10
133/133 [==============================] - 2573s 19s/step - loss: 0.2108 - accuracy: 0.9894 - val_loss: 7.9421 - val_accuracy: 0.9492
Epoch 7/10
133/133 [==============================] - 2571s 19s/step - loss: 0.2826 - accuracy: 0.9860 - val_loss: 4.3349e-08 - val_accuracy: 0.8983
Epoch 8/10
133/133 [==============================] - 2571s 19s/step - loss: 0.0849 - accuracy: 0.9934 - val_loss: 0.8763 - val_accuracy: 0.9322
Epoch 9/10
133/133 [==============================] - 2561s 19s/step - loss: 0.1382 - accuracy: 0.9931 - val_loss: 2.4481 - val_accuracy: 0.9492
Epoch 10/10
133/133 [==============================] - 2547s 19s/step - loss: 0.0951 - accuracy: 0.9943 - val_loss: 2.4979 - val_accuracy: 0.9407
```

Figure 4.18: Accuracy Diagram for ResNetV2-152.

Confusion matrix:

States the accuracy of identification. Here, Accurate identification of Amloki, Ata, Bel and Orbori fruit.

```
Confusion Matrix
[[ 8  0  0  0  0  0  0]
 [ 0 18  0  0  0  0  0]
 [ 3  0 17  0  0  0  0]
 [ 0  0  0 26  0  0  0]
 [ 0  0  0  0 15  0  1]
 [ 0  0  0  0  0 14  0]
 [ 1  0  0  2  0  0 13]]
```

Figure 4.19: Confusion Matrix for ResNetV2-152.

Classification Report:

Shows precision, recall, f1-scrore and accuracy between the fruit classes. Achieved an accuracy rate of 94% from this model.

Classification Report

	precision	recall	f1-score	support
amloki	0.67	1.00	0.80	8
ata	1.00	1.00	1.00	18
bangi	1.00	0.85	0.92	20
bel	0.93	1.00	0.96	26
chalta	1.00	0.94	0.97	16
orbori	1.00	1.00	1.00	14
sofeda	0.93	0.81	0.87	16
accuracy			0.94	118
macro avg	0.93	0.94	0.93	118
weighted avg	0.95	0.94	0.94	118

Figure 4.20: Classification Report for ResNetV2-152.

Train Accuracy Vs Val Accuracy:

Noticeable difference between both curves.

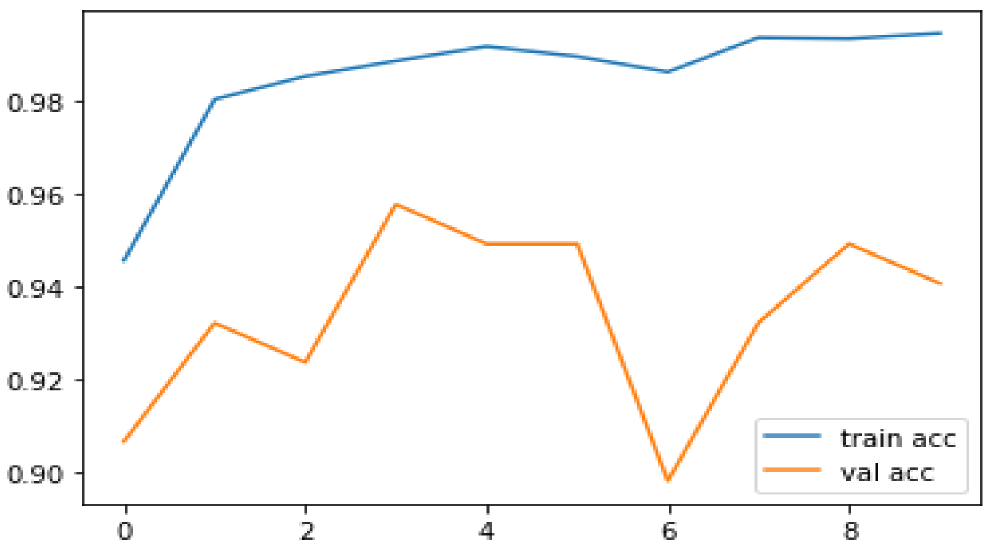


Figure 4.21: Train Accuracy vs Validation Accuracy for ResNetV2-152

Train Loss Vs Val Loss:

Noticeable fluctuations on the validation loss curve.

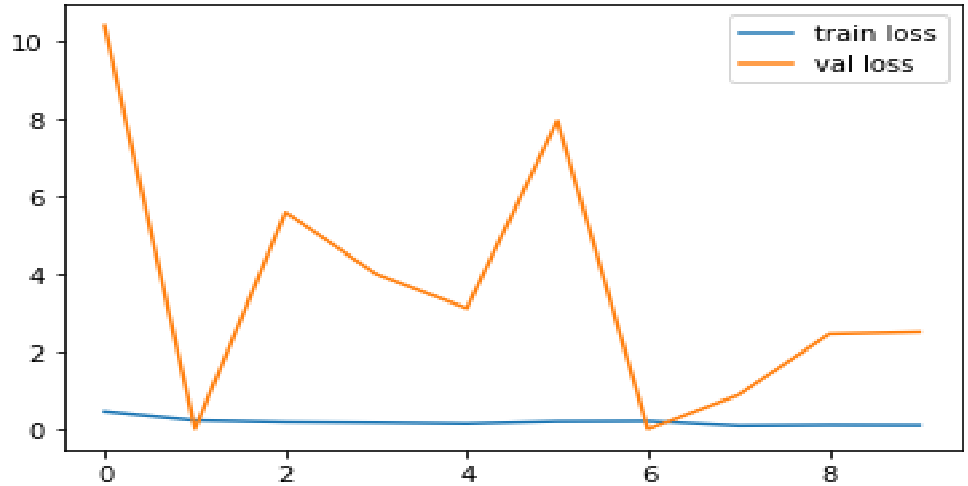


Figure 4.22: Train Loss vs Validation Loss for ResNetV2-152.

4.6 Experimental Result and Analysis for Xception Model:

Parameter:

Total parameters: 21,563,951.

Trainable parameters: 702,471.

Non-trainable parameters: 20,861,480.

Accuracy:

Accuracy stated for each number of epochs. Here, Increasing the number of epochs results in inconsistent accuracy.

```
Epoch 1/10
133/133 [==============================] - 1675s 13s/step - loss: 0.6289 - accuracy: 0.9068 - val_loss: 4.4423 - val_accuracy: 0.8220
Epoch 2/10
133/133 [==============================] - 1616s 12s/step - loss: 0.4560 - accuracy: 0.9562 - val_loss: 15.3590 - val_accuracy: 0.7373
Epoch 3/10
133/133 [==============================] - 1614s 12s/step - loss: 0.2213 - accuracy: 0.9733 - val_loss: 11.0331 - val_accuracy: 0.7627
Epoch 4/10
133/133 [==============================] - 1624s 12s/step - loss: 0.4857 - accuracy: 0.9664 - val_loss: 35.3464 - val_accuracy: 0.6949
Epoch 5/10
133/133 [==============================] - 1618s 12s/step - loss: 0.2035 - accuracy: 0.9785 - val_loss: 29.7311 - val_accuracy: 0.6780
Epoch 6/10
133/133 [==============================] - 1609s 12s/step - loss: 0.0753 - accuracy: 0.9891 - val_loss: 16.1132 - val_accuracy: 0.7288
Epoch 7/10
133/133 [==============================] - 1614s 12s/step - loss: 0.1336 - accuracy: 0.9860 - val_loss: 22.0848 - val_accuracy: 0.7373
Epoch 8/10
133/133 [==============================] - 1611s 12s/step - loss: 0.0948 - accuracy: 0.9886 - val_loss: 26.5599 - val_accuracy: 0.7119
Epoch 9/10
133/133 [==============================] - 1613s 12s/step - loss: 0.3349 - accuracy: 0.9775 - val_loss: 16.4230 - val_accuracy: 0.7881
Epoch 10/10
133/133 [==============================] - 1614s 12s/step - loss: 0.1344 - accuracy: 0.9882 - val_loss: 53.0129 - val_accuracy: 0.6949
```

Figure 4.23: Accuracy Diagram for Xception.

Confusion matrix:

States the accuracy of identification. Here, Accurate identification of Chalta and Orbori fruit.

```
Confusion Matrix
[[ 7  0  0  0  0  1  0]
 [ 0 13  0  1  1  3  0]
 [ 0  0 17  2  0  1  0]
 [ 4  0  0 14  0  8  0]
 [ 0  0  0  0 16  0  0]
 [ 0  0  0  0  0 14  0]
 [ 0  0  0  1  0 14  1]]
```

Figure 4.24: Confusion Matrix for Xception.

Classification Report:

Shows precision, recall, f1-scrore and accuracy between the fruit classes. Achieved an accuracy rate of 69% from this model.

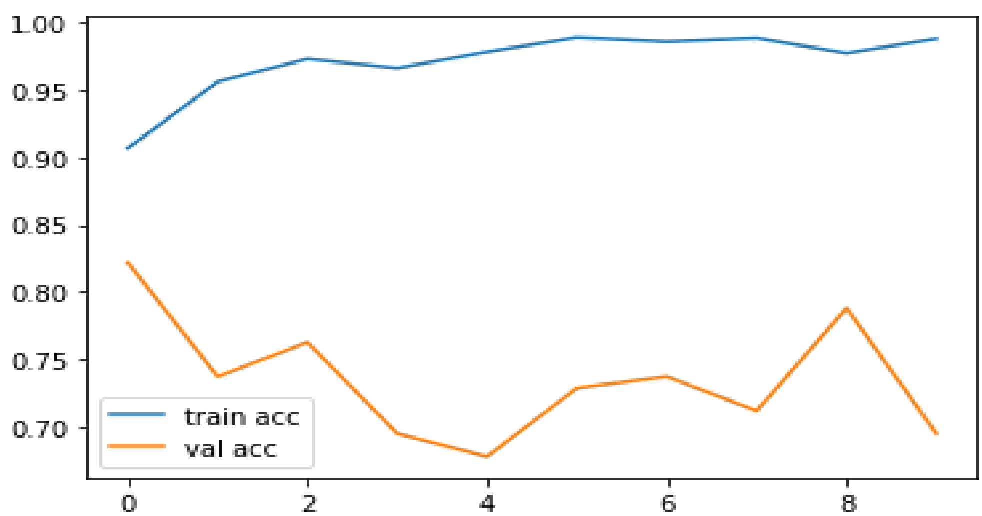
Classification Report

	precision	recall	f1-score	support
amloki	0.64	0.88	0.74	8
ata	1.00	0.72	0.84	18
bangi	1.00	0.85	0.92	20
bel	0.78	0.54	0.64	26
chalta	0.94	1.00	0.97	16
orbori	0.34	1.00	0.51	14
sofeda	1.00	0.06	0.12	16
accuracy			0.69	118
macro avg	0.81	0.72	0.68	118
weighted avg	0.84	0.69	0.68	118

Figure 4.25: Classification Report for Xception.

Train Accuracy Vs Val Accuracy:

Immense difference between the curves is noticeable.



:

Figure 4.26: Train Accuracy vs Validation Accuracy for Xception.

Train Loss Vs Val Loss:

Gradual increment of validation loss is noticeable whereas train loss remained constant.

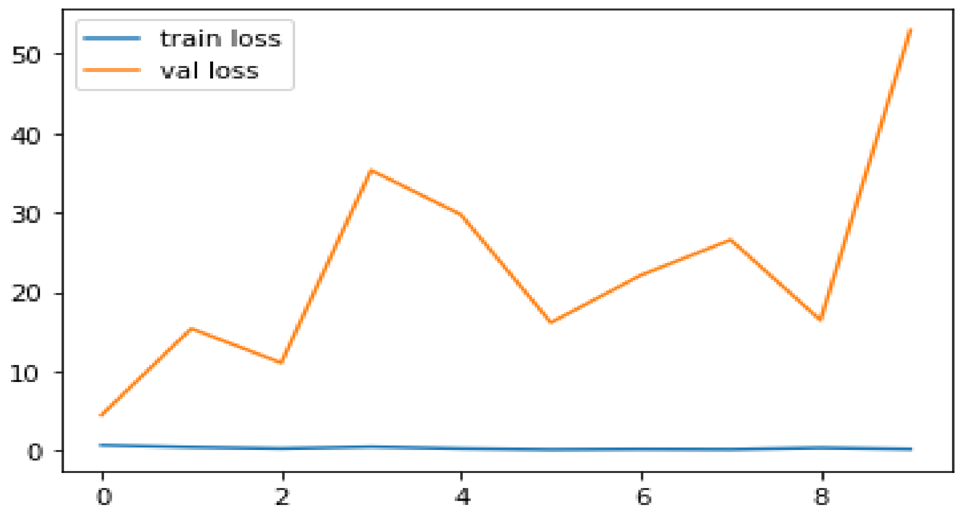


Figure 4.27: Train Loss vs Validation Loss for Xception.

4.7 Discussion:

From the results achieved from testing the DenseNet-201 Model [14] it is evident that as we increase the number of epochs, the difference between the train accuracy and the validation accuracy increases. If we evaluate the confusion matrix for this model, we will notice that it accurately identified Amloki and Orbori fruit. It falsely predicted 11 out of 18 Ata fruit for Bel fruit, 1 out of 20 Bangi fruit for Bel fruit, 3 out of 26 bel fruit for Sofeda fruit, 6 out of 16 chalta fruit for Bel fruit and finally, 2 Amloki and 1 Bel fruit for 16 Sofeda fruit respectively. We have achieved an accuracy rate of 80% from this model.

For InceptionV3 Model [8], The curves between train accuracy and validation accuracy were getting closer. Evaluating the confusion matrix, we notice that, the model was able to accurately identify Amloki, Ata, Bangi, Bel, Chalta, Orbori fruits respectively. It only falsely identified the Sofeda fruit for 4 Bel and 10 Orbori fruits respectively. We have achieved an accuracy rate of 86% from this model.

Upon analyzing the VGG-16 Model [9], The curves between the train accuracy and the validation accuracy was considerable closer. Evaluating the confusion matrix, we notice that, the model was able to accurately identify all of the fruits. We have achieved an accuracy rate of 100% from this model.

Analyzing the ResNetV2-152 Model [13], we have noticed that train loss and validation loss curves fluctuated heavily. This model accurately identified only Amloki, Ata, Bel and Orbori fruit. We have achieved an accuracy rate of 94% from this model.

Analyzing the Xception Model [12], We have noticed a large difference between the curves of train accuracy and validation accuracy from the very beginning of the epochs. In the confusion matrix we notice that, this model was only able to accurately identify Chalta and Orbori fruit. We have achieved an accuracy rate of 69% from this model.

Based on the results achieved it is evident that the VGG-16 Model performed best with an accuracy rate of 100%.

CHAPTER 5
IMPACT ON SOCIETY, ENVIRONMENT AND SUSTAINABILITY

5.1 Impact on Society

Using DCNN for fruit recognition has massive impact on society. Industries are using this architecture in automated crop collecting robots. Farmers can use this technique for yield estimation too. This could generate healthy growth of crops as well as more harvest throughout the year. Farmers no longer need to depend on environmental limitations.

5.2 Impact on Environment

Agricultural robotics depend gravely on crops classifications and identifications. With deep learning techniques it is now possible for detecting fruit diseases in a large cluster. It is helping the healthy growing of crops and it can be used in large greenhouses which is beneficial for the environment.

5.3 Ethical Aspects

These classifications techniques can be implemented without harming the environment rather it can be helpful for it. Classification of fruits could generate larger crop yields without any wastage this could be helpful for eliminating the use of artificial ripening of the fruits.

5.4 Sustainability Plan

Smart mobile applications can be developed for instant identification of fruits that are native to a specific region or get important details like nutritional values etc. Super shops can use these applications for tracking stocks too.

CHAPTER 6
CONCLUSION, RECOMMENDATION AND IMPLICATION FOR FUTURE RESEARCH

6.1 Conclusions:

In our research project we compared and evaluated five powerful deep learning models like Xception, Inception, VGG-16, ResNet and Densenet for recognition of fruits that are native to our country Bangladesh. We also discussed their working principals and applications. Among these five models VGG-16 deep learning model has shown its exquisite potential for recognition of fruits. We have established our own database from 7 different classes of fruit for experimentation. After preprocessing the data, we trained and tested all five models. Our goal was to test the efficiency and accuracy of all five models regarding fruit recognition.

6.2 Recommendations:

The performance of the deep learning models we discussed depend on various factors however, all five of them are capable of fruit detection. On the basis of the results and accuracy with our own dataset we recommend VGG-16 [9] model. Compared to all four of the models VGG-16 model provided highest accuracy rate of 100% on our dataset.

6.3 Implication for Future Research:

For further studies in the future, we can include more classes of fruits and focus on sub-class categorization. We can also introduce larger datasets as most DCNN show clear accuracy rates when tested with large datasets. Since, these models are capable of detecting objects from images regardless of the type, other food items and vegetables can also be used for training. We can also include various real-life challenges like different lighting conditions and clustered positions of fruits. This will involve further experimentation of the structure of the convolutional neural networks.

References:

[1] Hou, L., Wu, Q., Sun, Q., Yang, H., Li, P. (2016). FRUIT RECOGNITION BASED ON CONVOLUTIONAL NEURAL NETWORK, 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD), Changsha, pp. 18-22.

[2] Sa, I., Ge, Z., Dayoub, F., Upcroft, B., Perez, T., & McCool, C. (2016). A FRUIT DETECTION SYSTEM USING DEEP NEURAL NETWORKS, Sensors (Basel) 2016 Aug; 16(8): 1222. Published online 2016 Aug 3. doi: 10.3390/s16081222.

[3] Hussain, I., HE, Q., & Chen, Z. (2018). AUTOMATIC FRUIT RECOGNITION BASED ON DCNN FOR COMMERCIAL SOURCE TRACE SYSTEM, International Journal on Computational Science & Applications (IJCSA) Vol.8, No.2-3.

[4] Shamim Hossain, M., Al-Hammadi, M., & Muhammad, G. (2018). AUTOMATIC FRUIT CLASSIFICATION USING DEEP LEARNING FOR INDUSTRIAL APPLICATIONS, IEEE Transections on Industrial Informatics, Volume: 15, Issue: 2, 1027-1034.

[5] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. IMAGENET LARGE SCALE VISUAL RECOGNITION CHALLENGE. 2014.

[6] G. Hinton, O. Vinyals, and J. Dean. DISTILLING THE KNOWLEDGE IN A NEURAL NETWORK, 2015.

[7] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. GOING DEEPER WITH CONVOLUTIONS. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 1–9, 2015.

[8] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. RETHINKING THE INCEPTION ARCHITECTURE FOR COMPUTER VISION. arXiv preprint arXiv:1512.00567, 2015.

[9] K. Simonyan and A. Zisserman. VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE IMAGE RECOGNITION. arXiv preprint arXiv:1409.1556, 2014.

[10] Yamamoto, K.; Guo, W.; Yoshioka, Y.; Ninomiya, S. ON PLANT DETECTION OF INTACT TOMATO FRUITS USING IMAGE ANALYSIS AND MACHINE LEARNING METHODS. Sensors 2014, 14, 12191–12206.

[11] Wang, Q.; Nuske, S.T.; Bergerman, M.; Singh, S. AUTOMATED CROP YIELD ESTIMATION FOR APPLE ORCHARDS. In Proceedings of the 13th Internation Symposium on Experimental Robotics (ISER 2012), Québec City, QC, Canada, 17–22 June 2012.

[12] Collet, F. (2017). XCEPTION: DEEP LEARNING WITH DEPTHWISE SEPARABLE CONVOLUTIONS, 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), DOI: 10.1109/CVPR.2017.195.

[13] Kaiming He et al. (2016). DEEP RESIDUAL LEARNING FOR IMAGE RECOGNITION, Computer vision and pattern recognition, pp.770-778.

[14] Huang, G., Liu, Z., & Maaten, L.V.D. (2017). DENSELY CONNECTED CONVOLUTIONAL NETWORKS, 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), arXiv:1608.06993.

[15] Yanai, K., & Kawan, Y. (2015). FOOD IMAGE RECOGNITION USING DEEP CONVOLUTIONAL NETWORK WITH PRE-TRAINING AND FINE-TUNING, 2015 IEEE International Conference on Multimedia & Expo Workshops (ICMEW), Turin, Italy.

[16] Tammina, S. (2019). TRANSFER LEARNING USING VGG-16 WITH DEEP CONVOLUTIONAL NEURAL NETWORK FOR CLASSIFYING IMAGES, International Journal of Scientific and Research Publications, Volume 9, Issue 10, DOI: 10.29322/IJSRP.9.10. 2019.p9420.

[17] Zeng, G. (2017). FRUIT AND VEGETABLES CLASSIFICATION SYSTEM USING IMAGE SALIENCY AND CONVOLUTIONAL NEURAL NETWORK, IEEE 3rd Information Technology and Mechatronics Engineering Conference (ITOEC), Chongqing, pp. 613-6.

Plagiarism Report

ORIGINALITY REPORT

10%	5%	6%	6%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Daffodil International University Student Paper	2%
2	Bini Alias, R Karthika, Latha Parameswaran. "Classification of High Resolution Remote Sensing Images using Deep Learning Techniques", 2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2018 Publication	1%
3	Lei Hou, QingXiang Wu, Qiyan Sun, Heng Yang, Pengfei Li. "Fruit recognition based on convolution neural network", 2016 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD), 2016 Publication	1%
4	neurohive.io Internet Source	1%
5	Submitted to Pioneer Academics Student Paper	1%