A MINI PROJECT REPORT

on

"Fruit classification using deep learning approach"

Submitted by

Ashwin George Mathew	4NM19IS037
Adithya	4NM19IS009
Amith Krishna	4NM19IS022

BACHELOR OF ENGINEERING

in

INFORMATION SCIENCE AND ENGINEERING

Under the Guidance of

Mr. Abhishek S. Rao

Asst. Professor

Department of Information Science and Engineering

In partial fulfillment of the requirements for the award of

Bachelor of Engineering in Information Science andEngineering

From

Visvesvaraya Technological University,

BelagaviNitte - 574 110, Karnataka, India

May-2022

CERTIFICATE

This is to certify that the mini project entitled **"Fruit classification using deep learning approach"** has been carried out by **Ashwin George Mathew (4NM19IS037), Adithya (4NM19IS009), and Amith Krishna (4NM19IS022)** the bonafide students of NMAM Institute of Technology, Nitte in Information Science and Engineering of Visvesvaraya Technological University, Belagavi during the year 2021 - 2022. It is certified that all corrections/suggestions indicated for internal assessment have been incorporated in the report deposited in the departmental library. The mini project report has been approved as it satisfies the academic requirements in respect of mini project work prescribed for the said degree in the sixth semester.

Signature of the Guide	**Signature of the Coordinator**	**Signature of the HOD**
Mr. Abhishek S. Rao	Mr. Vasudeva Pai	Dr . Karthik Pai B H

External Viva:

Examiner's Name	Signature with Date
1.	. . . .
2.	. . . .

DECLARATION

We hereby declare that the entire work embodied in this Mini Project Report titled **"Fruit classification using deep learning approach"** has been carried out by us at NMAM Institute of Technology, Nitte under the supervision of **Mr. Abhishek S. Rao,** for **Bachelor of Engineering** in **Information Science and Engineering**. This report has not been submitted to this or any other University for the award of any other degree.

Ashwin George Mathew(4NM19IS037)

Adithya(4NM19IS009)

Amith Krishna(4NM19IS022)

Dept. of ISE, NMAMIT, Nitte

Abstract

Recent advances in computer vision have enabled a wide range of applications in all areas, leaving agriculture untouched. The use of advanced technology is essential for the food industry. With the ability of deep learning to learn robust features from images, we have found tremendous applications in multiple disciplines. Due to the shape, color and texture of different types of fruits, identifying and classifying fruits remains a challenge. Until 2018, many traditional machine learning techniques were used to investigate the impact of computer vision on fruit detection and classification, but some techniques use deep learning techniques for fruit detection and classification. The propose study has led to extensive research on the collection and implementation of deep learning models for fruit recognition and classification. This article details the challenges of detecting and classifying fruits using datasets, practical descriptors, model implementations, and deep learning used by many scientists. In addition, the popular Fruit 360 dataset was used to implement a deep learning fruit classification model from scratch to help novice farmers understand the role of deep learning in agriculture.

Acknowledgement

It is with great satisfaction and euphoria that we are submitting the Mini Project Report on **“Fruit classification using deep learning approach”**. We have completed it as a part of the curriculum of Visvesvaraya Technological University, Belagavi for the award of Bachelor ofEngineering in Information Science and Engineering.

We are profoundly indebted to our guide, **Mr.Abhishek S.Rao** Assistant Professor, Department of Information Science and Engineering for innumerable acts of timely advice, encouragement and We sincerely express our gratitude.

We express our sincere gratitude to **Dr. Karthik Pai B H**, Professor and Head, Department of Information Science and Engineering for his invaluable support and guidance.

We sincerely thank **Dr. Niranjan N Chiplunkar**, Principal, NMAM Institute of Technology, Nitte and **Dr. I Ramesh Mithanthaya**, Vice Principal & Dean (Academics),NMAM Institute of Technology, Nitte, who have always been a great source of inspiration.

Finally, yet importantly, we express our heartfelt thanks to our family and friends for theirwishes and encouragement throughout the work.

Ashwin George

Mathew(4NM19IS037)

Adithya(4NM19IS009)

Amith Krishna(4NM19IS022)

Table of Contents

CHAPTER 1

Introduction

India is one of the largest fruit-producing countries. Identifying defective fruit and classifying fruit as "fresh" or "rotten" is posing a major problem in the Indian agricultural industry. Since it is difficult in the industry to classify fruit quality by conventional methods, a new technology based on image processing is needed for fruit classification. Manual, human-led reviews are often error-prone, inconsistent, and time-consuming. The Machine Vision system can be used to perform automatic classification. This avoids the problems associated with the traditional classification. Recent advances in Deep Learning, especially in the field of computer vision, have improved the efficiency of classification. This method is more efficient, error-free, and enables automated image-based inspection analysis. Figure Classification methods based on artificial vision are becoming more and more popular.

Image classification is a very active research direction in many fields and plays a very important role. Image recognition serves many uses, including video analysis, face recognition, image classification, and more. Deep learning (DL) is a subfield of machine learning (ML) that has shown excellent results in image recognition. DL uses a multi-layer structure to process image features, greatly increasing the efficiency of image recognition. In other words, DL and image recognition applications are becoming a concept in logistics and supply chains

The current development of computer vision (CV) has shown outstanding results in many areas of life. The detection and classification of fruit have proven to be a complex and difficult task. For certain economic sectors, both the wholesale and retail market, the study of fruit processing is very important, including the processing industry. These factors have prompted researchers to develop different methods for automated processing of fruits, to identify them or estimate their quality efficiently. In recent years, agricultural industries such as food processing, marketing, packaging, and fruit grading have become a more focused research direction. The processing and grading of single crops such as oranges, cherries, apples, mangoes, and citrus fruits are laborious and time-consuming because there are many varieties of the same fruit, for example, more than 7,000 varieties of apples are produced worldwide.
Automation can thus reduce labor costs and rapidly increase productivity. In the initial study, the scientists proposed different methods from the CV for manual extraction of fruit characteristics and ML for classification of CV characteristics. Through CV algorithms, the characteristics of fruit color, shape, size, and texture are used for classification algorithms.

Chapter 2

Problem Definition

2.1 Objectives

i. To collect the dataset from the Kaggle repository

ii. To preprocess the data for better performance of the model

iii. To design the classification model by using a deep learning approach

iv. To corroborate the model for its efficacy by using various performance metrics

2.2 Problem Definition

Human-made manual assessments are error-prone, inconsistent, and time-consuming. The Machine Vision system can be used to perform automatic classification. CNN and VGG 16

Chapter 3

Software Requirements Specification

3.1 Project requirement:

1. TensorFlow library
2. Keras Library
3. Spyder IDE
4.Fuit 360 Dataset

3.2 Hardware requirement:

1.Processor - minimum Intel(R) Core(TM) i5 processor

2. Processor speed - minimum 1.5 GHz

3. RAM - minimum 8.00 GB

4. RADEON Graphics Card 2GB

3.3 Software requirement:

1. Anaconda framework

Anaconda is a distribution of the Python programming language for scientific computing (data science, machine learning applications, large-scale data processing, predictive analytics, etc.), that aims to simplify package management and deployment.

2. Spyder

Spyder is an open-source cross-platform integrated development environment (IDE) for scientific programming in the Python language. Spyder integrates with a number of prominent packages in the scientific Python stack, including NumPy, SciPy, Matplotlib, and pandas.

3. Windows OS

Windows is a group of several proprietary graphical operating system families, all of which are developed and marketed by Microsoft.

LIBRARIES

1.Tensor Flow

TensorFlow is an end-to-end open source platform for machine learning. It has a comprehensive, flexible ecosystem of tools, libraries, and community resources.

2. Keras

Keras is an API designed for human beings, not machines. Keras follows best practices for reducing cognitive load: it offers consistent & simple APIs

3. ImageDataGenerator

Keras ImageDataGenerator class provides a quick and easy way to augment your images. It provides a host of different augmentation techniques like standardization, rotation, shifts, flips, brightness change, and many more.

4. Numpy

NumPy is a popular image processing Python library with ndarray to set and modify pixel values, trim images, concatenate images, and many more. One can complete multiple image processing without using other Python libraries. It helps in colour reduction, binarization, paste with slice, positive or negative inversion, and many more functionalities for efficient reading and saving images.

5. Pandas

The Library is written in Python Web framwork and is used for data manipulation for numerical data and time series. It uses data frames and series to define three-dimensional and two-dimensional data respectively. It also provides options for indexing large data for quick search in large datasets. It is well known for the capabilities of data reshaping, pivoting on user-defined axis, handling missing data, merging and joining datasets, and the options for data filtrations. Pandas is very useful and very fast with large datasets..

6. Matplotlib

Matplotlib is an amazing visualization library in Python for 2D plots of arrays. Matplotlib is a multi-platform data visualization library built on NumPy arrays and designed to work with the broader SciPy stack.

Matplotlib is very helpful for being an image module to work with images in Python. It also includes two useful methods that are used to read images as well as to display the image. It is specialized in 2D plots of arrays as a multi-platform data visualization library on Numpy arrays.

Chapter 4

Methodology

Convolution Neural Network

Convolution Neural Networks are neural networks that share their parameters. Convolution layers consist of a set of learnable filters as shown fig 1. Every filter has small width and height and the same depth as that of input volume (3 if the input layer is image input).

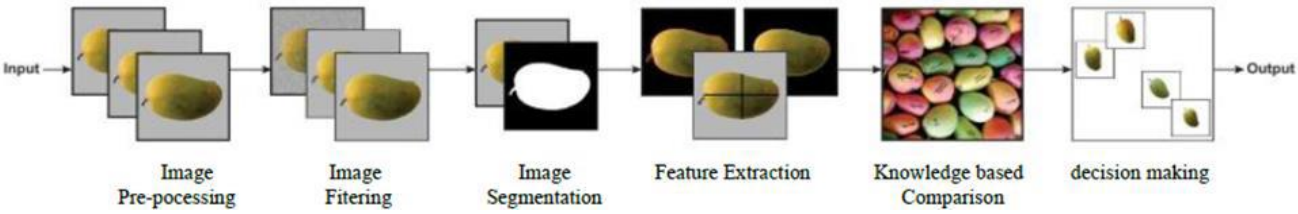

Fig 1. The flow of fruit classification process

a. Image Pre-processing & Filtering

This section removes noise, sharp, smoothens the image also performs resizing of images. RGB images are converted to grey images also the contrast of images is increased at a certain level. Such preprocessing operations are also named filtration.

b. Image Segmentation

Segmentation is used for partitioning an image into various parts. The goal of image segmentation is to simplify and/or change the representation of an image, which is more meaningful and easier to analyze. Image segmentation methods are categorized on the basis of two properties of discontinuity and similarity. Methods based on discontinuities are called boundary-based methods, and methods based on similarity are called region-based methods.

c. Feature Extraction

The next step in the fruit classification process after segmentation is feature extraction. The main and important visual external features of fruit are its color, size, shape, and texture. Feature descriptor is a representation of an image or part of it, which extracts useful information and discards unnecessary information. It is mainly used for image recognition and object detection.

d. Knowledge-based Comparison and Decision Making

The comparison of the extracted features from the image takes place with the predetermined classification and sorting criteria or the rules. The features are compared on the basis of the extracted features and classification is made are given to the fruits. The knowledge-based comparison and decision-making has been made using Convolutional neural network algorithms. This is the final step of the classification system

Convolution process

During the forward pass, we slide each filter across the whole input volume step by step where each step is called stride computes the dot product between the weights of filters and patch from input volume. As we slide our filters we'll get a 2-D output for each filter and we'll stack them together and as a result, we'll get output volume having a depth equal to the number of filters. The network will learn all the filters.
In order to get a better accuracy, we will use a VGG-16 model which consists of 16 layers

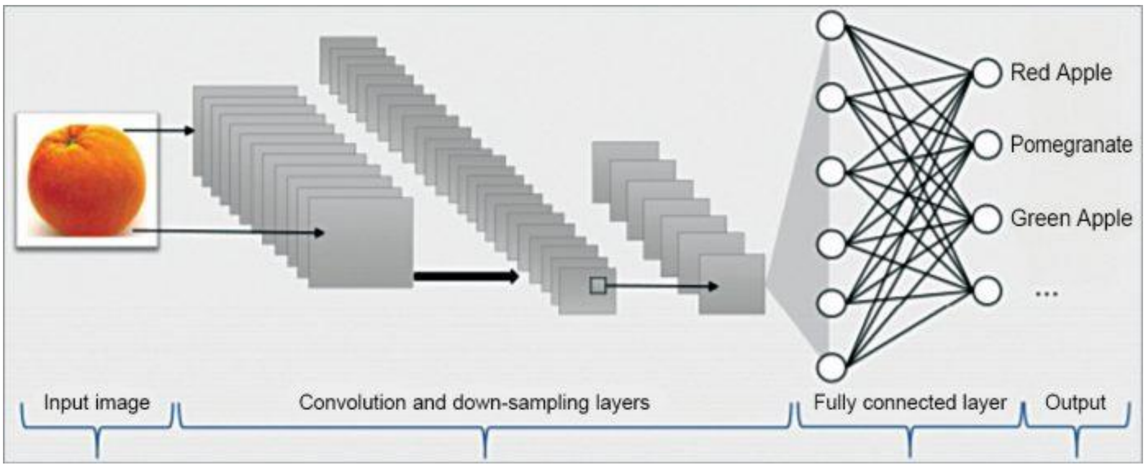

Fig 2 CNN layers

VGG-16

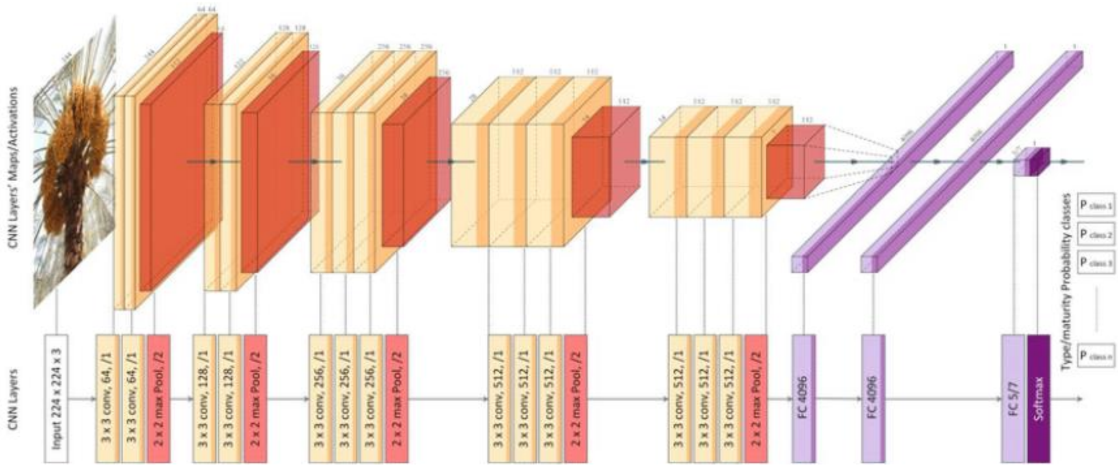

Fig 3. VGG16 layers

VGG-16 which is a 16-layer convolution model as show in fig 3. It was compared by Bargoti and Underwood against a faster region-based convolutional neural network for deep fruit detection in orchards. The performance from the convolution layers is a map with high dimensions, sampled by 16 due to the steps of the layers in the pooling. In the local function map areas, a layer of box regression and a box classification layer is distributed to two entirely related siblings

Chapter 5

Implementation and Results

Dataset

For training and testing, all the images were selected from the fruits-360 dataset which is publicly available on Github and Kaggle. The dataset contains 65,429 different fruit images of 95 categories.

We picked around 7000 images from 11 different categories. Among them to create the training set we used 80%images and the rest 20% of images were used for testing the model. The dataset of 11 different categories of fruits as shown in fig4.

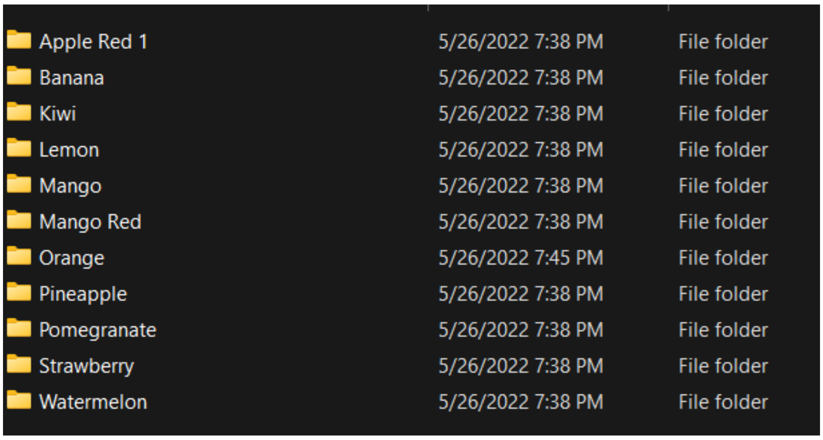

Fig4. datasets

Initially, we import the required packages. if these packages are not installed then the installation is done. Then an image is selected at random from the dataset samples. Then Sanity check is done to check if the path is set and the image is picked.

The VGG16 pre-trained model has 16 layers. we use activation as softmax as it has multiple classes. For binary classification we use sigmoid.

ImageDataGenerator is to import data with labels easily into the model. The output layer is then flattened and prediction is done using softmax. Flatten is used to flatten the multi-dimensional input tensors into a single dimension.

Then the compling is done using 'rmsprop' optimizer. The model summary is then generated as shown in fig 5.1 and fig 5.2.

Model Summary :

```
In [66]: model.summary()
Model: "functional_3"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_2 (InputLayer)         [(None, 100, 100, 3)]     0
_________________________________________________________________
block1_conv1 (Conv2D)        (None, 100, 100, 64)      1792
_________________________________________________________________
block1_conv2 (Conv2D)        (None, 100, 100, 64)      36928
_________________________________________________________________
block1_pool (MaxPooling2D)   (None, 50, 50, 64)        0
_________________________________________________________________
block2_conv1 (Conv2D)        (None, 50, 50, 128)       73856
_________________________________________________________________
block2_conv2 (Conv2D)        (None, 50, 50, 128)       147584
_________________________________________________________________
block2_pool (MaxPooling2D)   (None, 25, 25, 128)       0
_________________________________________________________________
block3_conv1 (Conv2D)        (None, 25, 25, 256)       295168
_________________________________________________________________
block3_conv2 (Conv2D)        (None, 25, 25, 256)       590080
_________________________________________________________________
block3_conv3 (Conv2D)        (None, 25, 25, 256)       590080
_________________________________________________________________
block3_pool (MaxPooling2D)   (None, 12, 12, 256)       0
_________________________________________________________________
block4_conv1 (Conv2D)        (None, 12, 12, 512)       1180160
_________________________________________________________________
block4_conv2 (Conv2D)        (None, 12, 12, 512)       2359808
_________________________________________________________________
block4_conv3 (Conv2D)        (None, 12, 12, 512)       2359808
```

Fig5.1 model summary

```
_________________________________________________________________
block4_pool (MaxPooling2D)   (None, 6, 6, 512)         0
_________________________________________________________________
block5_conv1 (Conv2D)        (None, 6, 6, 512)         2359808
_________________________________________________________________
block5_conv2 (Conv2D)        (None, 6, 6, 512)         2359808
_________________________________________________________________
block5_conv3 (Conv2D)        (None, 6, 6, 512)         2359808
_________________________________________________________________
block5_pool (MaxPooling2D)   (None, 3, 3, 512)         0
_________________________________________________________________
flatten_1 (Flatten)          (None, 4608)              0
_________________________________________________________________
dense_1 (Dense)              (None, 11)                50699
=================================================================
Total params: 14,765,387
Trainable params: 50,699
Non-trainable params: 14,714,688
_________________________________________________________________
```

Fig5.2 model summary

A model summary is created when running the classification model. The model summary displays the name of the model, the model type, and the model formula. It gives total parameters, trainable parameters, and nontrainable parameters and layers used.

We use model.fit_generator as we are using ImageDataGenerator to pass data to the model. We will pass train and test data to fit_generator. In fit_generator steps_per_epoch will set the batch size to pass training data to the model and validation_steps will do the same for test data. We can tweak it based on the system specifications.

Once we have trained the model we can visualize training/validation accuracy and loss. We pass the output of mode.fit_generator to the history variable. All the training/validation accuracy and loss are stored in history and we will visualize it from there.

We can visualize the epoch loss and the epoch accuracy with the help of a matplotlib Library.

Epoch loss:

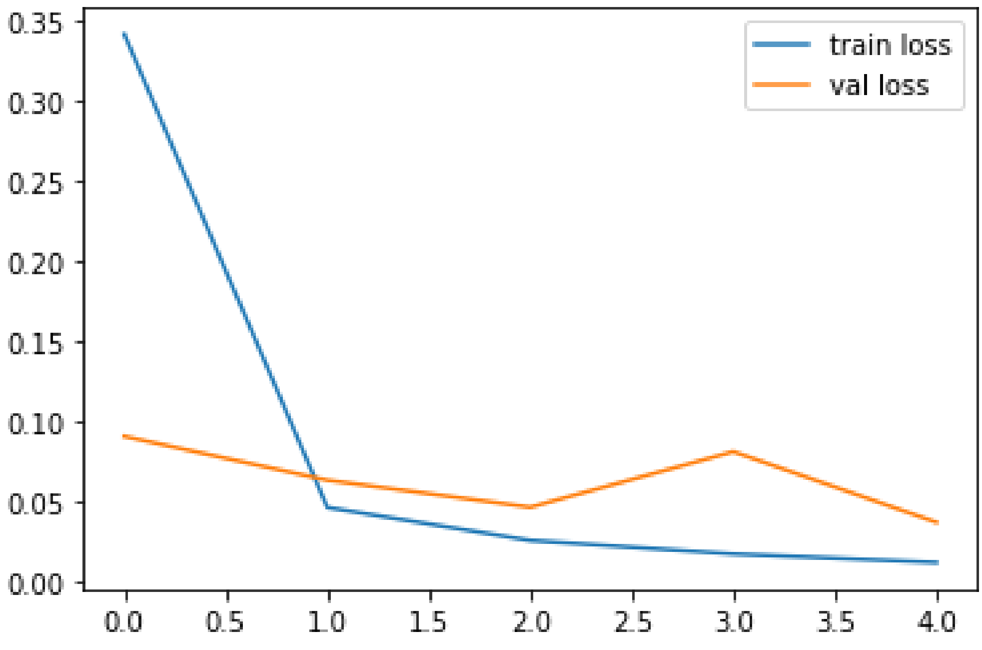

Fig6.1 Epoch loss

To do the prediction the image is loaded onto the model and the model predicts the result as the output.

Epoch accuracy:

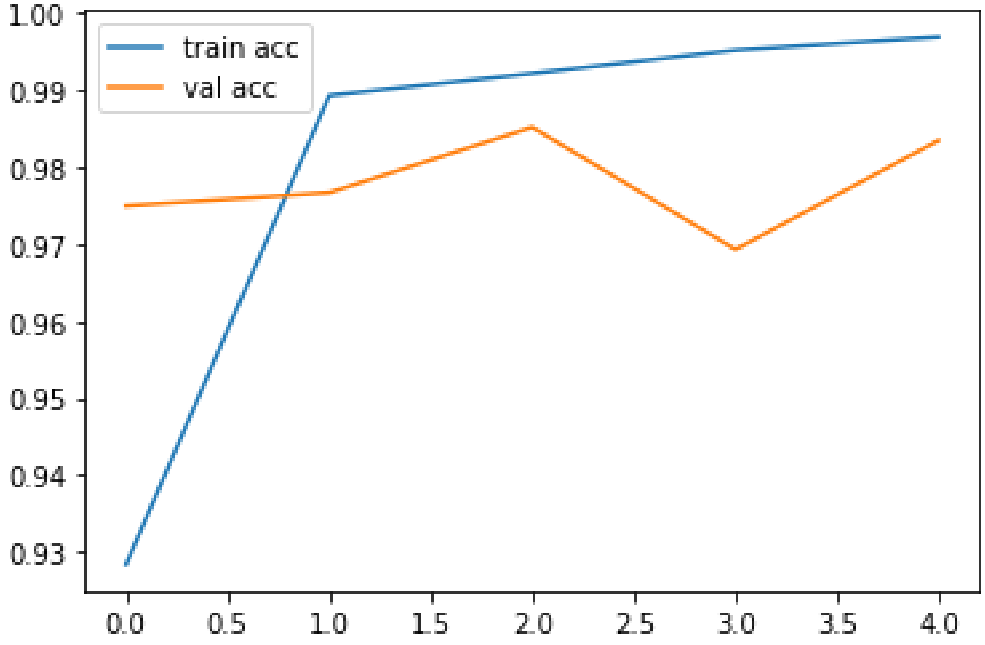

Fig6.2 Epoch accuracy

RESULTS

```
In [64]: from IPython.display import Image, display
    ...: for i in range(0, 3):
    ...:   display(Image(test_files[i]))
    ...:   print("Actual class: " + str(actual_res[i]))
    ...:   print("Predicted class: " + str(test_res[i]))
Actual class: Strawberry\61_100.jpg
Predicted class: Strawberry
Actual class: Strawberry\53_100.jpg
Predicted class: Strawberry
Actual class: Mango\293_100.jpg
Predicted class: Mango
```

Fig7.1 output 1

```
In [65]: result = np.round(model.predict(valid_generator))
    ...: import random
    ...: test_files = []
    ...: actual_res = []
    ...: test_res = []
    ...: valid_path=valid_path+'/'
    ...: for i in range(0, 3):
    ...:   rng = random.randint(0, len(valid_generator.filenames))
    ...:   test_files.append(valid_path+  valid_generator.filenames[rng])
    ...:   actual_res.append(valid_generator.filenames[rng].split('/')[0])
    ...:   test_res.append(labels[np.argmax(result[rng])])
    ...:
    ...: from IPython.display import Image, display
    ...: for i in range(0, 3):
    ...:   display(Image(test_files[i]))
    ...:   print("Actual class: " + str(actual_res[i]))
    ...:   print("Predicted class: " + str(test_res[i]))
Actual class: Banana\294_100.jpg
Predicted class: Banana
Actual class: Banana\296_100.jpg
Predicted class: Banana
Actual class: Banana\304_100.jpg
Predicted class: Banana
```

Fig7.1 output 2

Chapter 6

Conclusion

This study investigated and analyzed using various deep learning methods proposed by many researchers in the field of fruit detection and classification. After reviewing various automated approaches to fruit recognition and classification, we found that previous reviews focused on the application of computer vision technology in this area. However, despite the cutting-edge performance, deep learning models have received less attention due to many image classification issues. To fill this gap, we used a deep learning model to perform a recent review of recently published literature in the field of fruit detection and classification.

In addition, detailed studies were presented that considered feature descriptions, algorithms for recognition and classification, and various datasets for fruit recognition and classification. In addition, after a critical analysis of the reviewed method, open issues related to datasets, feature representations, and classification algorithms need to be identified and addressed. In addition, in order to deepen the use of the DL model in the agricultural field, we conducted an experiment using the CNN model

References

[1] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, et al., "Backpropagation applied to handwritten zip code recognition," Neural computation, vol. 1, pp. 541-551, 1989.

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

[3] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 1725-1732.

[4] Y. Sun, Y. Chen, X. Wang, and X. Tang, "Deep learning face representation by joint identification-verification," in Advances in neural information processing systems, 2014, pp. 1988-1996.

[5] F. Siddique, S. Sakib, and M. A. B. Siddique, "Handwritten Digit Recognition using Convolutional Neural Network in Python with Tensorflow and Observe the Variation of Accuracies for Various Hidden Layers," 2019

[6] Xiaoyang Liu, Dean Zhao,Weikuan jia, Wei Ji, Yueping Sun, “A Detection Method for Apple Fruits Based on Color and Shape Features” , IEEE Access, 22 May 2019 .

[7] Seng, Woo Chaw, and Seyed Hadi Mirisaee. "A new method for fruits recognition system." Electrical Engineering and Informatics 2009. ICEEI'09. International Conference on. Vol. 1. IEEE, 2009.

[8] Nishat Tasnim, Md. Romyull Islam, and Shaon Bhatta Shuvo “A Convolution Neural Network Based Classification Approach for Recognizing Traditional Foods of Bangladesh from Food Images”.