Fruit Quality Determination using Image Processing and Deep Learning

A Dissertation submitted

for the partial fulfillment of the degree of

Bachelor of Engineering in

Information Technology

(Session 2022 -2023)

Guided By:

Mr. C. P. Patidar

Submitted By:

Lakshya Gour (19I7124)
Ritik Kumar Koshta (19I7139)
Udbhav Gupta (19I7165)

Department of Information Technology

Institute of Engineering & Technology

Devi Ahilya Vishwavidyalaya, Indore (M.P.)

(www.iet.dauniv.ac.in)

December 2022

Dissertation Approval Sheet

The dissertation entitled **"Fruit Quality Determination using Image Processing and Deep Learning"** submitted by **Lakshya Gour, Ritik Kumar Koshta and Udbhav Gupta** is approved as partial fulfillment for the award of **Bachelor of Engineering Information Technology** degree by **Devi Ahilya Vishwavidyalaya, Indore**.

Internal Examiner **External Examiner**

Director

Institute of Engineering & Technology

Devi Ahilya Vishwavidyalaya,

Indore (M.P.)

Recommendation

The dissertation entitled **"Fruit Quality Determination using Image Processing and Deep Learning"** submitted by **Lakshya Gour, Ritik Kumar Koshta and Udbhav Gupta** is a satisfactory account of the bonafide work done under my supervision is recommended towards the partial fulfillment for the award of **Bachelor of Engineering in Information Technology** degree by **Devi Ahilya Vishwavidyalaya, Indore.**

Date: / /

Project Guide

Mr. C. P. Patidar

Endorsed By

Head, Department of Information Technology

Candidate Declaration

We hereby declare that the work which is being presented in this project "**Fruit Quality Determination using Image Processing and Deep Learning"** in partial fulfillment of degree of Bachelor of Engineering in **Information Technology** is an authentic record of our own work carried out under the supervision and guidance of **Mr. C. P. Patidar**, **Assistant Professor** in Department of **Information Technology**, Institute of Engineering and Technology, Devi Ahilya Vishwavidyalaya, Indore

We are fully responsible for the matter embodied in this project in case of any discrepancy found in the project and the project has not been submitted for the award of any other degree.

Date: / /

Place: Indore

Lakshya Gour (19I7124)

Ritik Kumar Koshta (19I7139)

Udbhav Gupta (19I7165)

ACKNOWLEDGEMENTS

Words can never express the extent of indebtness but we still wish to express our deep sense of gratitude to all the people who helped us in the completion of this project.

I want to express my heart-felt gratitude to **Mr. C. P. Patidar** for his advises and unremitting support over the last one year. He trained me how to do research and how to write, encouraged me not to be intimidated by the difficult problems. He is so generous with his time and ideas. His intellectual creativity, perseverance and commitment would benefit me for the rest of my life. I could never thank him enough for being an excellent mentor and a wonderful teacher.

My thanks also go to **Dr. Vrinda Tokekar, HOD IT Department,** for the discussion during the course of the project, for the reading of the draft of this thesis and providing valuable feedback.

I want to thank the director of our college **Dr. Sanjiv Tokekar** for assisting us. Therefore, my thanks go to them for making this project possible.

I also thank my team members and friends for supporting me and helping me out with the testing phase as well as rest of the project.

Finally, I want to thank my parents for giving me strength and love.

ABSTRACT

A considerably high amount of fruit produced is wasted due to improper management and utilization during harvesting, storing, transporting, and in the food processing industry. Fruit will get rotten easily if not stored properly due to bacteria accumulation. It is known to all that rotten or defective fruits are harmful to health. It may damage the fresh fruits which are in surface contact with the rotten fruits in the inventory. These rotten fruits should be detected and sorted as early as possible. The problem that comes across in manual checking by humans is less uniformity and accuracy as the manual examination by humans' eye will consume time and energy. This research proposes a method involving the deep learning technique which is CNN (Convolutional Neural Networks) for feature extraction and classification of rotten fruits. It is one of the applications of image classification problems. This approach uses an RGB channel image of the fruit under examination. The image will be evaluated by the trained model as Fresh if the percentage of rotten part detected is under the threshold value. The types of fruits that will be identified and classified in this paper are apple, banana and orange. Transfer learning technique is used, which minimizes training time and resources and aids to achieve higher accuracy. The dataset is divided into two parts, for (70%) training and (30%) validation. The raw image set used for training is first pre-processed and then fed into the model.The validation accuracy obtained in this paper is 98.47%. The total duration of the training stage is 210.37 minutes. Hence, the required time to classify a single fruit image is approximately 0.2 second. Our model can be adopted by industries closely related to the fruit cultivation and retailing or processing chain for automatic fruit identification and classifications in the future.

TABLE OF CONTENTS Page No

Chapter-1
Introduction

1.1 Overview and Issues Involved

Fruits, being highly rich in nutrition, are one of the most widely consumed foods as a part of the recommended daily diet. A major chunk of fruit produced worldwide is used by the food processing industry. Fruits have a long way to travel from the moment it is grown by farmers till it reaches the end consumers or until it is processed in companies to produce various products. During its lifespan, it highly possesses a risk of getting rotten. Rottenness is the state of decomposition or decay of the quality of the fruit, which not only affects the taste and appearance but also alter its nutritional composition, causing the presence of mycotoxins dangerous for humans. According to a journal of medical science, about 2 million deaths happen every year in India due to consuming contaminated food and water. According to a report published by the World Health Organisation (WHO), over half a million deaths occur every year globally due to unsafe food, where children under five are most affected. In India, households have an alarming 13.2 percent prevalence of hazardous food practices.

1.2 Problem Defination

Automatic fruit classification is an interesting problem in the fruit growing and retailing industrious chain because it can help the fruit growers and supermarkets identify different fruits and their status from the stock or containers so as to improve production efficiency and hence business profit. It will also help in reducing the consumption of stale fruits.

Presently, rottenness identification is carried out using human scrutiny or using Ultraviolet light to highlight spots of rottenness represented as fluorescence. These manual methods for segmentation of fruits are either slow and less accurate, require large human efforts or may expose fruits to harmful radiation.

Automatic detection of rotten fruits can alleviate the costs of the manual and ad-hoc filtering activity. The state of the art in this topic reports the capturing of fruit images and then, use of deep learning algorithms on neural networks for processing those images. Among the various types of deep learning neural networks, CNNs are the best suited for understanding image pixels because its built-in convolutional layer reduces the high dimensionality of images without losing its information.

Since, the input is in the form of image of the fruit, texture: color and size are the significant parameters for fruit freshness determination. The rotten as well as defected fruits are well distinguished using color. Analyzing the fruit quality which is based on color, shape and size should be done in an economic and a safe way. Testing should be done by non-destructive techniques because these are delicate materials.. It is also helpful in planning, packaging, transportation and marketing operations. If the distinction is done manually, the process will be too slow and prone to error. Humans classify the fruits on the basis of color, shape, size. If these quality parameters are mapped into an automated fruit grading and detection system by using suitable programming language then the work will be faster and hassle free.

1.3 Proposed Solution

1.3.1 Proposed Method

The core of our ML (Machine Learning) model is the CNNs of Inception v3. We have used transfer learning approach, which will ease the real life implementation of our solution. Inorder to completely automate the task of filtering unusable fruits at large scale, the ML model can be deployed on a backend using framework such as Django which will be hosted over cloud services such as AWS or Azure. Snapshots of the fruits coming over conveyor will be captured continuously. APIs will be built using RESTFUL architecture which will be used for sending these images to the backend for detection and server will return a response in Json or XML format. Bots, based on the results from our model, will filter out the stale fruits.

In order to validate the proposal, a web based user interface is built for user where they just need to upload image of the fruits and then at server there will be a model which will be trained on dataset and which will detect the freshness of fruits and return the result which will be displayed on the webpage. The backend where the ML model will be deployed is built using django, which is a python based backend framework. HTML, CSS, Javascript is used in creating frontend User Interface.

A brief summary of the major product functions and what the end user may perform on the application include:

● User opens the web application in the browser.

● User will upload an image of a fruit by clicking on the upload image button.

● After uploading the image the user will click on the result button to get results.

● Then the result will be returned from the server and the user can view the result on the webpage.

1.3.2 Workflow

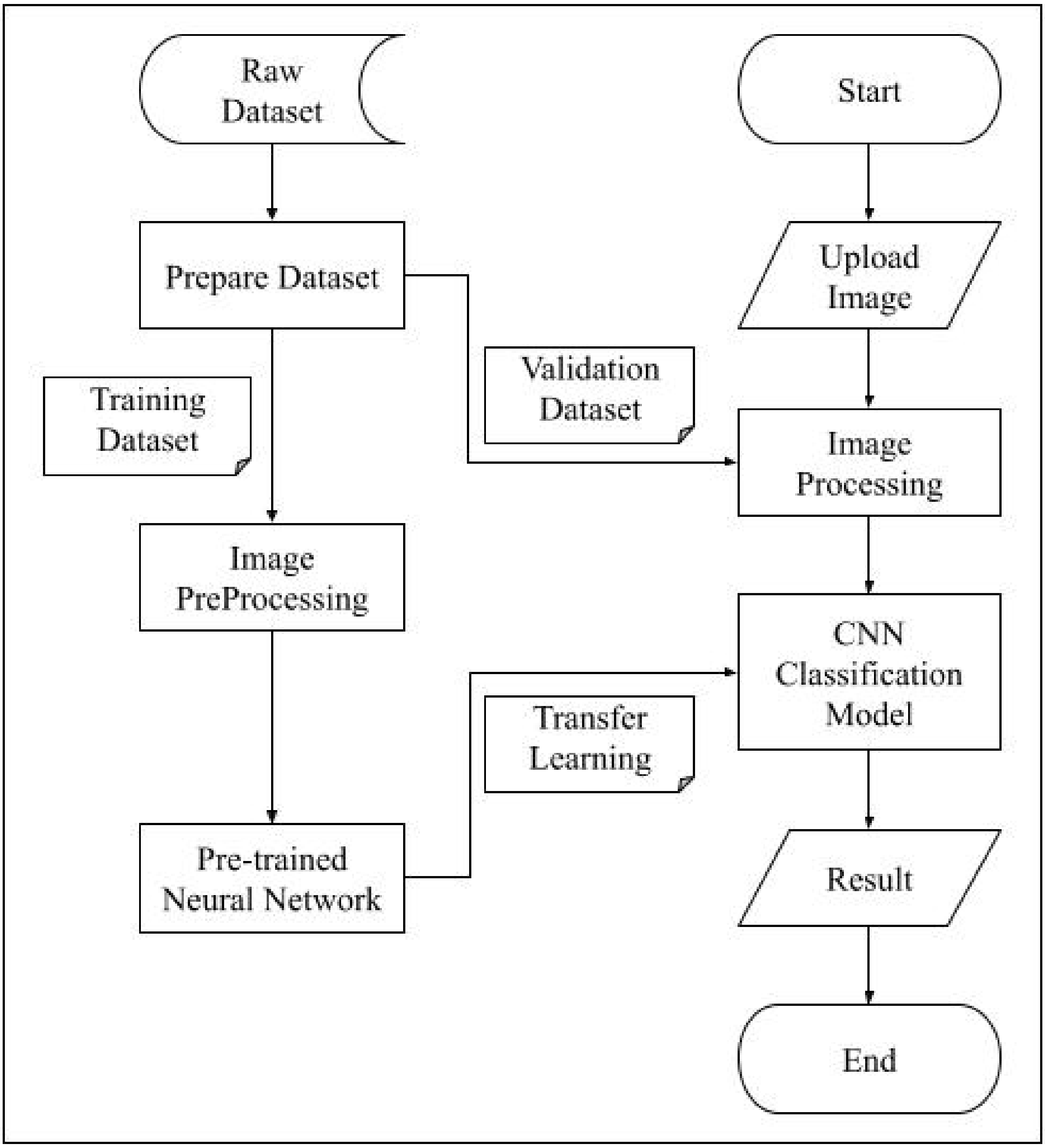

Fig. 1. Workflow of our proposed methodology

Chapter-2

Literature Survey

2.1 Existing Solutions

Deep learning has been revolutionized and used in many fields such as agriculture, healthcare and so on. There is a lot of research going on about deep learning.

Wang, L., Li, A., Tian, X. used a machine vision system in their study to detect defects in the skin of the fruit. The main feature and support used in color for classification is a ML algorithm called Vector Machine (SVM) used in classification and as a recognizer. Support vector machines (SVMs) give perfect and suitable results for a fewer number of datasets. The accuracy of the classification method using ML mostly depends on the drawn features. And the features are selected to go into the ML algorithm [5].

D. Karakaya, O. Ulucan and M. Turkan in their work, have investigated the performance of multiple feature extraction techniques on fruit freshness classification. After the feature extraction of fruits images, the classifier technique which is again support vector machine (SVM) was used in the experiment. A total of 1200 fruits images were used where the images are categorized into three different fruits classes such as orange, banana and apple. Among all the feature extraction techniques, CNNsF (Convolutional Neural Networks Features) achieved the best performance with an overall accuracy of 97.61%. The performance of the experiment is obtained by computing the success rates of different SVM features while each class is trained with a one-vs-all SVM classifier per feature [6].

In another work, A. Wajid, N. K. Singh, P. Junjun and M. A. Mughal had investigated the performance of multiple types of classifiers which are Decision Tree, Artificial Neural Network (ANN) and Naïve Bayes on distinguishing the condition of orange. The condition is categorized as ripe, unripe, scaled or rotten. The features of orange images such as RGB color space and gray values are extracted using BIC in this paper. After the comparison of multiple types of classifiers, Decision Tree has the highest accuracy which is 93.13%. The precision and sensitivity when using the Decision Tree classifier is 93.45% and 93.24% respectively.

The results are obtained by testing on a total of 335 orange images which include 125 unripe oranges, 85 ripe oranges and 125 scaled or rotten oranges [7].

In the work presented by Monika J. and Ashwani K, they had introduced the concept of artificial neural network (ANN) to detect disease and grading of fruit. The disease grading of fruits is divided into 5 by identifying the percentage of infection on fruits. Moreover, it computed the percentage of diseases for some part out of the total pixel counts of the fruits. After that, all the data computed is provided to ANN for training purposes. Back propagation algorithm is used to compute the output value in grading the query images which are not included in the learning database. The accuracy of the proposed method is 90% [8].

In the work done by Azizah, L.M.R., Umayah, S.F., Riyadi, S., Damarjati, C., Utama, the fruits are collected manually and the researchers themselves classify those fruits as fragile and defective. This is done in pre-processing images and given to the CNN model for classification. The accuracy of this model is 97.5%. Based on laser backscattering imaging analysis and CNN theory, this method gives a knowledge and theoretical basis for maximum productivity, non-destructive fruit quality identification. This work shows that the method is effective and non-destructive and can automatically identify defective areas. The effect of error detection based on the CNN model is better than conventional algorithms [14].

2.1.1 Disadvantages In These Methods

- Naïve Bayes, SVM does not perform very well when the data set has more noise i.e. target classes are overlapping.
- Classifier algorithms such as KNN, SVM, Naïve Bayes underperform compares to CNN.
- Large amount of dataset and more training time is required to train a CNN from scratch.

2.2 Transfer Learning Technique

Transfer learning approach is being used in the proposed method in order to make the model efficient in predicting if the fruit is rotten or fresh.

Transfer learning is a ML method where a learning model developed for a first learning task is reused as the starting point for a learning model in a second learning task

Often previous learning is referred to as source and the next learning as target. Basically it is using a pre-trained neural network (trained for Task1) for achieving shorter training time (positive transfer learning) in learning Task2. Using the conventional model for Image classification, we need lots of data to train the network. This task would require huge time, effort and cost which at many times, could make it practically infeasible. In this scenario, we can use the Transfer Learning technique, which uses pre-trained Neural Networks and uses the obtained weights on new data.

Fig. 2. describes the earlier layers of ConvNet that deal with generic features such as edge detectors, texture detectors, patterns. The later layers to the end of the network become more specific to the details of the image dataset. Here come the advantages of Transfer Learning. We will fix/freeze the earlier layers of the pre-trained network and retrain the rest of the layers with fine tuning of backpropagation. Instead of training our network from the first layer, we just transfer the weights already learned by a pre-trained network such as Inception v3 and save time and effort.

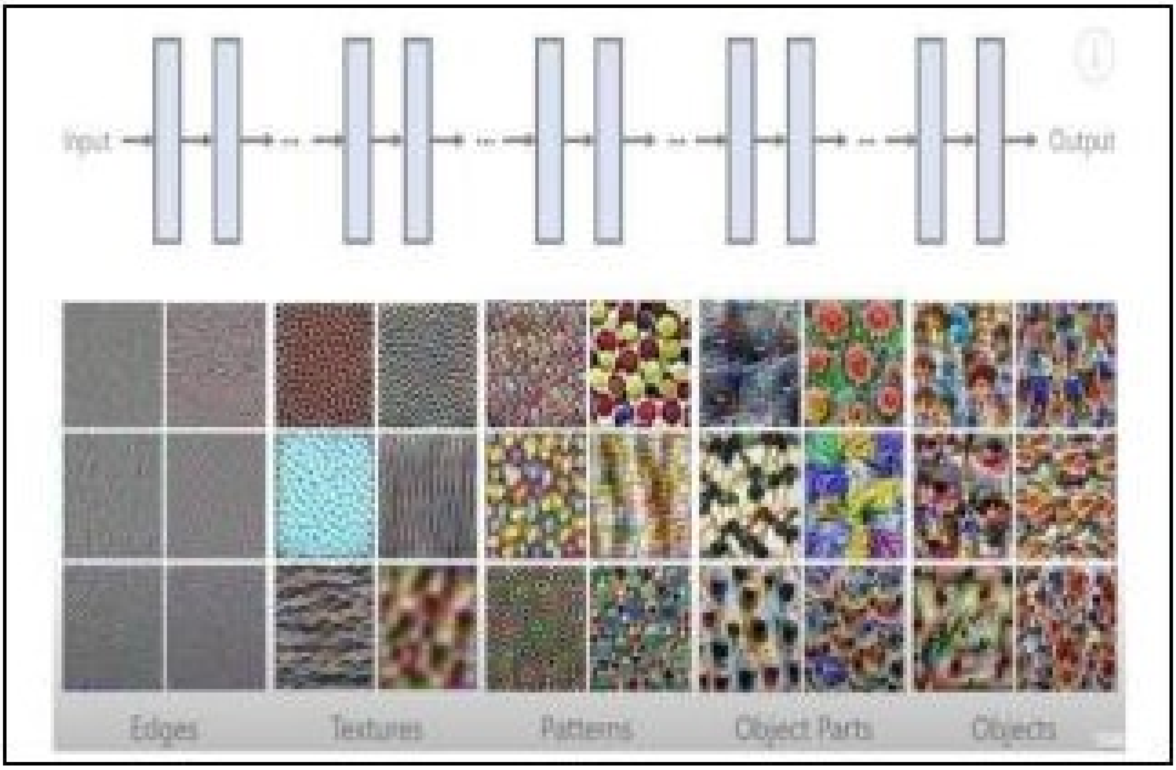

Fig. 2. Difference between Initial Layers and Later Layers [1]

Chapter-3
Methodology

3.1 Dataset Used

In our work, a smart fruit quality determination model is designed and developed which offers high accuracy. The types of fruits that will be focused in the analysis are banana, apple and orange. The experimental work is to classify whether the banana, orange and apple are fresh or rotten. In this proposed work, the algorithm that will be used is CNNs of Inception v3. All the images set are acquired from an open source website, namely Kaggle [10]. For experimenting with the dataset to perform rotten fruits classification, the training option parameter of mini batch size and the number of epochs in deep learning are studied to investigate and validate their relationship to the validation accuracy. The batch size is a hyperparameter that considers the number of examples from the training dataset used in the estimate of the error gradient before updating the model parameters. When the batch size is set to more than one and less than the examples in the training set, the learning algorithm is called mini-batch gradient descent. A considerably large number of fruit images is acquired to use as the dataset for training with CNNs.

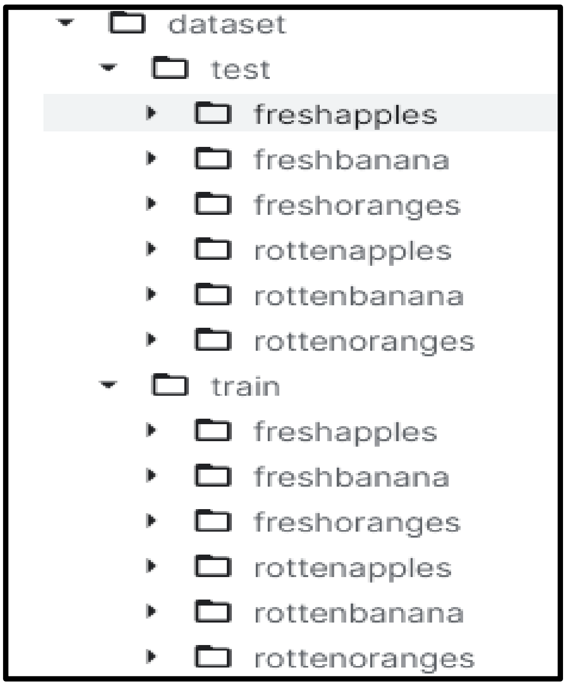

Fig. 3. Dataset Structure

TABLE I. Total number of fresh and rotten fruit images

Type of fruit	Training Set		Validation Set	
	Rotten	Fresh	Rotten	Fresh
Apple	350	350	150	150
Banana	350	350	150	150
Orange	350	350	150	150

The types of fruits to be classified are banana, apple and orange. Each of the fruits is categorized into fresh fruits and rotten fruits. The total amount of fruit images is 3000. 70% of the images are used for training dataset and 30% is for use as validation dataset. Table I shows the number of images as training and validation datasets for respective fruits. The training dataset is used to train the model, while the validation dataset is used to evaluate the model's performance.

3.2 Fundamental Proceses In Image Processing

There are some fundamental steps in image processing such as image acquisition, enhancement, and segmentation. Generally, the image acquisition stage involves some pre-processing steps, such as image rescaling, denoise, and smoothing edges. Image enhancement is among the most appealing areas in digital image processing. This process is to focus on the details of features that are blur and to improve them through the variation of brightness and contrast. While, the segmentation procedures partition an image into its constituent parts or objects [3]. In general, image segmentation is to change the representation of an image to become more detailed and easier to perform further processing. For example, it can remove the background of an image and take the foreground as the region of interest (ROI) for the next process. In this paper, the methodology in image processing is divided into 4 different stages as shown in Fig. 5. In the first stage, the input images which are the fruit images are loaded to the program to be processed. Then, the segmentation is performed on the images to extract the foreground of images. The foreground images will then perform with the feature extraction for classification in the system. Before processing the features extraction, all the images need to perform segmentation using the color thresholder function for foreground extraction on the images. The purpose is to reduce the processing time of the next process in feature extraction. The 'HSV' color space technique is used to detect the background of the fruit images. The representation of colors is hue (H), saturation (S) and value (V). Hue expresses the color from red to blue from the angle of 0 to 360 degrees. There are six different ranges of degrees which represent 6 different colors. Saturation controls the purity of color used. It is represented by the value ranges from 0 to 1. For the value of the model, it means the intensity or brightness of color [11].

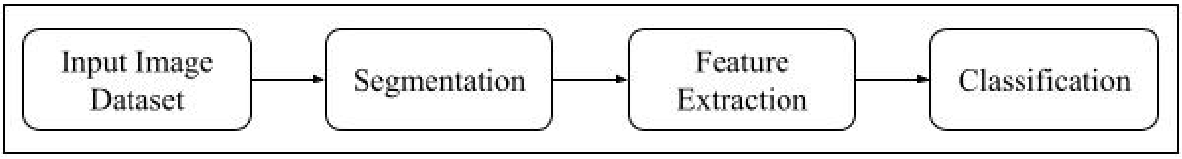

Fig. 4. General steps in Image Processing

3.3 Convolution Neural Network For Image Processing

In the field of deep learning, CNNs is considered as a class of deep neural networks. This deep learning technique is mainly used to analyze images for classification, object recognition, object detection and so on. The function of CNNs is to train and test an image. The images will go through multiple convolutional layers which are the filters, pooling, fully connected layers and lastly applying the softmax function to classify the input image. Fig. 5. shows the flow of CNNs from input image to classify the image based on the feature extracted. Convolution is the first layer to extract features from an input image. Convolution preserves the relationship between pixels by learning image features using small squares of input data. In other words, two inputs such as image and a filter are used to perform a mathematical operation. Sometimes, the filter used in the convolution layer does not fit perfectly with the input images. In this case, two options of padding are introduced. The first option is to pad the images with zeroes to make the pictures fit with the filter. Another option is valid padding which only take the valid part of images.

ReLU (Rectified Linear Unit) is the most common type of nonlinear activation function. The purpose of this function is to introduce non-linearity to the CNNs. This is because the real-world data may expect CNNs to learn non-negative linear values. Basically, this operation is to convert all the negative pixels of images to zero. When the size of images is too big, the pooling layer is used as a down sampling operation to reduce the spatial size while retaining the important parameters. The most commonly used type of pooling layer is max pooling which returns the largest value from the rectified feature regions.

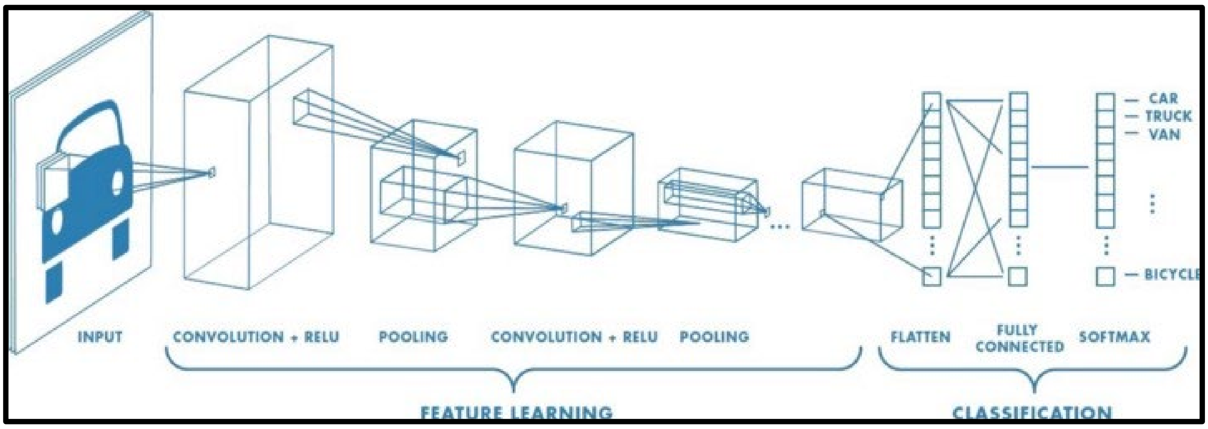

Fig. 5. Complete flow of CNNs [4].

3.4 Inception v3 Architecture

In a fully connected layer, all the features learned by the previous layer will be combined together to form a fully connected layer which is similar to a neural network. The purpose of this is for the system to identify the larger patterns of images. Finally, the activation function such as softmax function is used to classify the output [4]. As the number of layers increases, the capability for the CNNs to classify complex images also increases. In this paper, the features of images are extracted and classified using Inception v3. To reduce the computational time, Inception v3 is used as it is a pretrained CNN that consists of 48 layers deep. The pretrained network can classify images into 1000 object categories, such as keyboard, mouse, pencil, and animals [12]. Fig. 6. shows the model network structure of Inception v3. There is a large amount of feature representations extracted from the images that has been learned by these pretrained networks. However, to increase the accuracy on classifying the rotten fruits, the pretrained network is taken as a starting point to learn a new task. Instead of training a network from scratch, the pretrained network is modified with transfer learning. Another benefit of doing this is to reduce the required time of training the network. First, the pretrained network is loaded in MATLAB. All 48 layers of Inception v3 are loaded into a variable. The next step after loading the network is the replacement of the final connected layer. To modify this pretrained network to classify the rotten fruits, the fully connected (FC) layer in the pretrained network needs to be replaced by a new fully connected layer with the number of outputs equal to the number of classes to classify. In this case, the number of outputs is the layer. Normally, the last layer of the network which is the fully connected layer will be set as 6. In the network training stage, the size of images needs to be resized to meet the requirement of the pretrained network. For Inception v3, the size of input images is 299-by-299. Other than that, some training options need to be specified before the execution of training such as the learning rate factors, the number of epochs to train and the mini-batch size. The details of the training option are explained in the following section. Lastly, the validation accuracy is computed during the validation of the network.

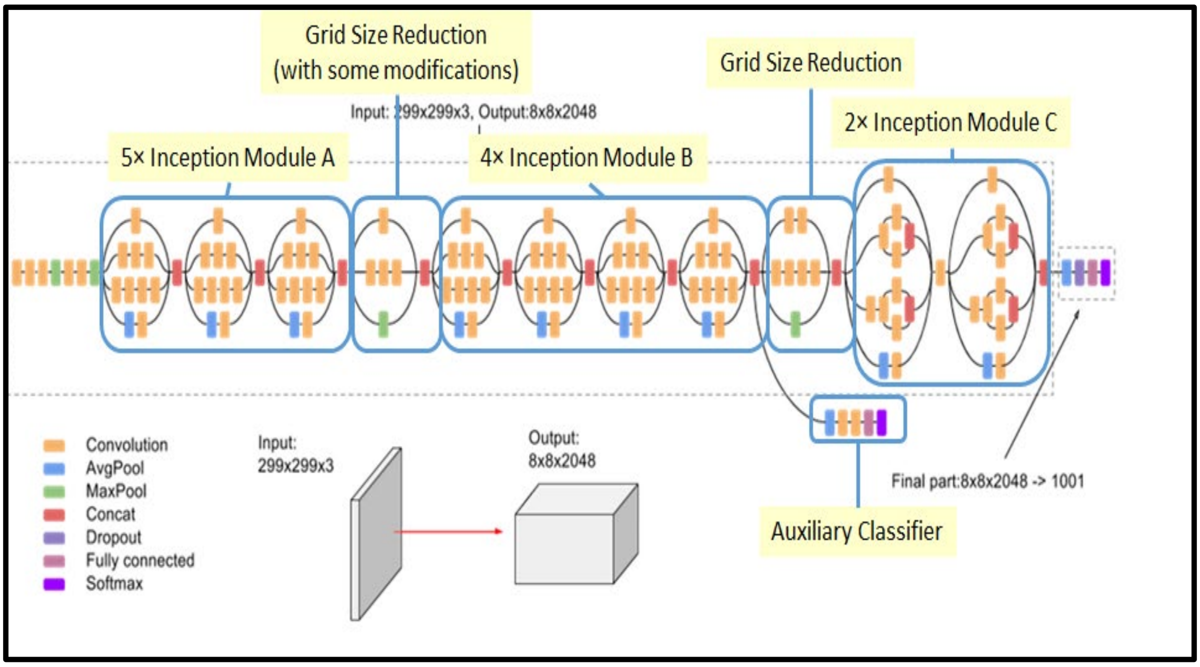

Fig. 6. Inception v3 model network structure [13]

3.5 Segmentation For Training

In this experiment, the segmentation process is performed on all the input images before the training stage. As shown in Fig. 7., it shows one of the rotten oranges with white fungus as the foreground pixels in the image. After segmentation, the foreground pixels from the image are not extracted out perfectly, leaving some part of the white fungus unsegmented correctly. The segmented image has the background of images' pixels reduced to zero value (black color).

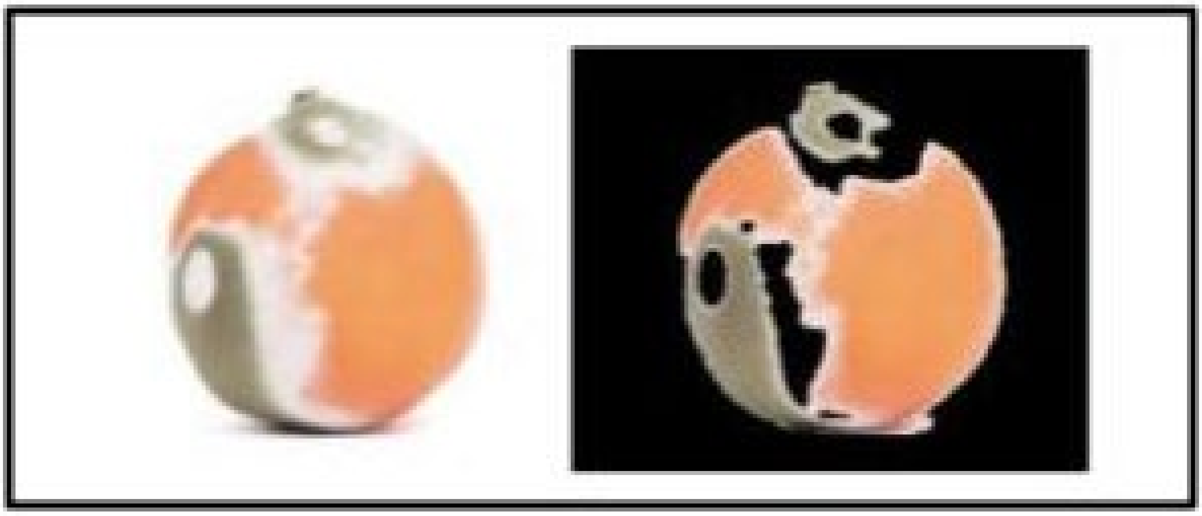

Fig. 7. Rotten orange with white fungus before segmentation (left) and after segmentation (right)

TABLE II. Comparison of accuracy before and after segmentation

Input Images	Mini-batch Size	Number of Epochs	Validation Accuracy	Training Time (minutes)
With Segmentation	10	6	98.47%	50.40
Without Segmentation	10	6	98.47%	44.72

Based on the table in Table II, the results show that the validation accuracy is the same for both cases which is 98.47%. It is observed that training the network using foreground detected images without segmentation has the similar accuracy with the fruit images after segmentation. The results also show the foreground detected images with segmentation require more training duration to train. Since all the original images are monotonous backgrounds in white and gray color, it is decided to use the original images without segmentation as the dataset for training purposes.

Chapter-4
Analysis And Design

4.1 Software Requirements

- **For Deep Learning**

  CNN is used for classification. The software requirements were python libraries and environment. The libraries used were Tensorflow, Keras, Matplotlib, Numpy. These libraries provided a wide range of functionalities.

  An online IDE for model training and validation is used : Jupyter Notebook

- **For Web Server**

  The web server relies on the Flask framework. Flask is a lightweight python framework which provides more secure and scalable features, also easy to integrate with our model.

- **For Client**

  The front end was developed using a javascript based framework : Svelte.

4.2 Hardware Requirements

- Minimum 2 GB RAM required.
- Internet Connection.
- Being a responsive website there is no constraints for the screen size of the device.

4.3 Analysis Diagrams

4.3.1 Use Case Model

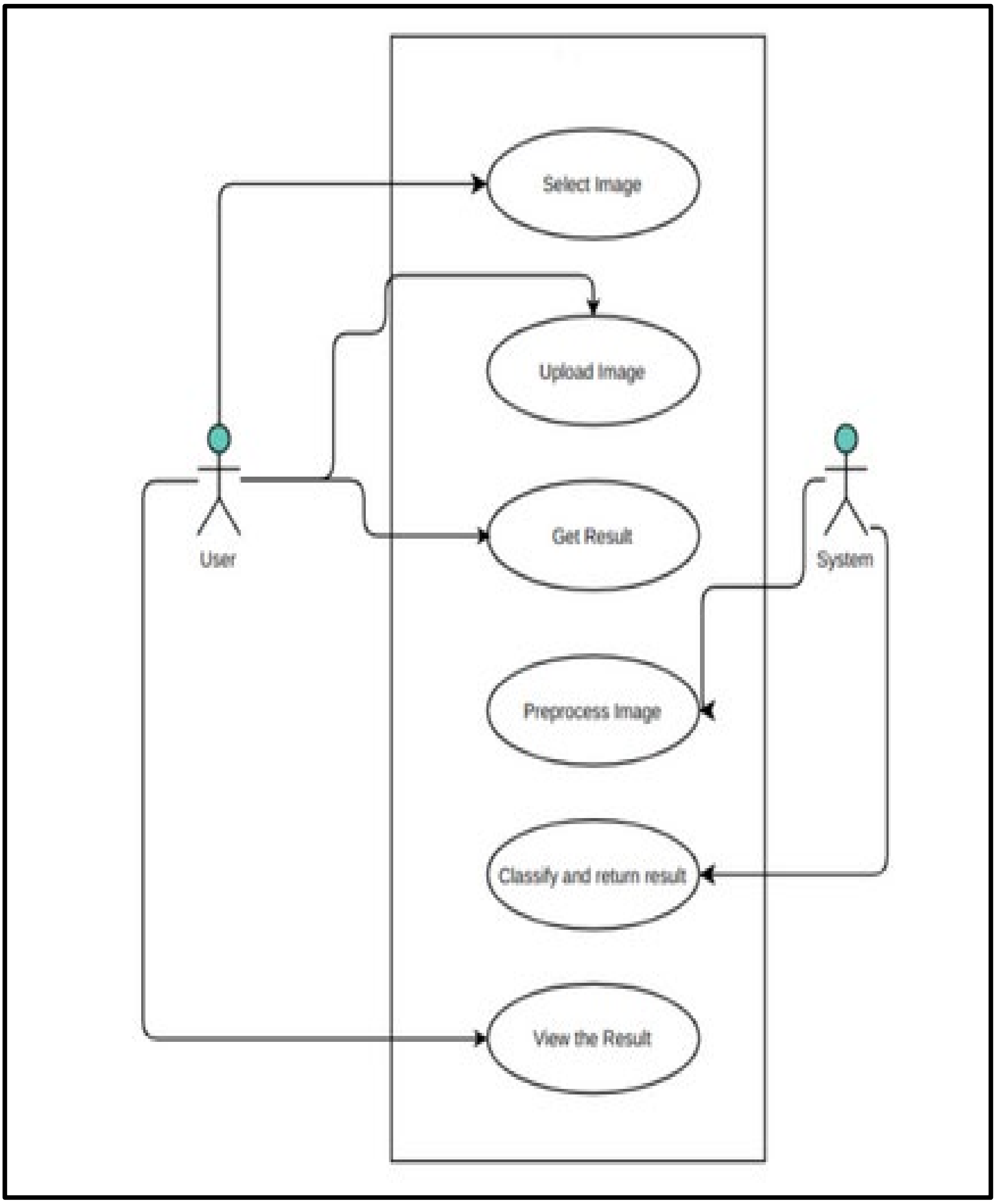

Fig. 8. Use Case Diagram

4.3.2 Use Case Description

- **Image Upload**

1 Brief Description -

This use case describes that the user can upload images.

2 Flow of Events -

- When the user visits the webpage of classification of images it sees upload button.
- He clicks on upload button to upload image by selecting it from dialog box.

3 Pre Conditions - None

4 Post Conditions -

Image gets successfully uploaded.

- **Select Image**

1 Brief Description -

This use case describes that the user can select an image.

2 Flow of Events -

- When the user clicks on the upload button it opens a dialog box to select an image.
- User selects the image and it gets uploaded.

3 Pre Conditions -

Image must be present.

4 Post Conditions -

Image gets successfully selected.

- **Get Result**

1 Brief Description -

This use case describes getting a result by clicking on a button.

2 Flow of Events -

- When the user clicks on the get result button the image is sent to the server.
- The server processes and return result.

3 Pre Conditions -

Image must be uploaded.

4 Post Conditions -

Successful request to get the result is sent.

- **Preprocess Image**

1 Brief Description -

This use case describes that the image will be preprocessed by the server.

2 Flow of Events -

- Images reach the server as the user clicks on the get result button.
- The server processes the image and modifies it into valid format and then passes it to the classification function.

3 Pre Conditions -

Image must be uploaded.

4 Post Conditions -

Image gets preprocessed successfully.

- **Classify and Return Result**

1 Brief Description -

This use case describes that the image will be used for classification and the result will be returned by the server.

2 Flow of Events -

- Image is passed to the model which classifies it.
- The server returns the result generated by the model.

3 Pre Conditions -

Image must be preprocessed.

4 Post Conditions -

Result must be returned in response.

- **View Result**

1 Brief Description -

This use case describes that the result can be viewed by the user.

2 Flow of Events -

When the result webpage opens the result will be displayed.

3 Pre Conditions -

Result must be generated**.**

4 Post Conditions -

Users can view the result.

4.4 Design Diagram

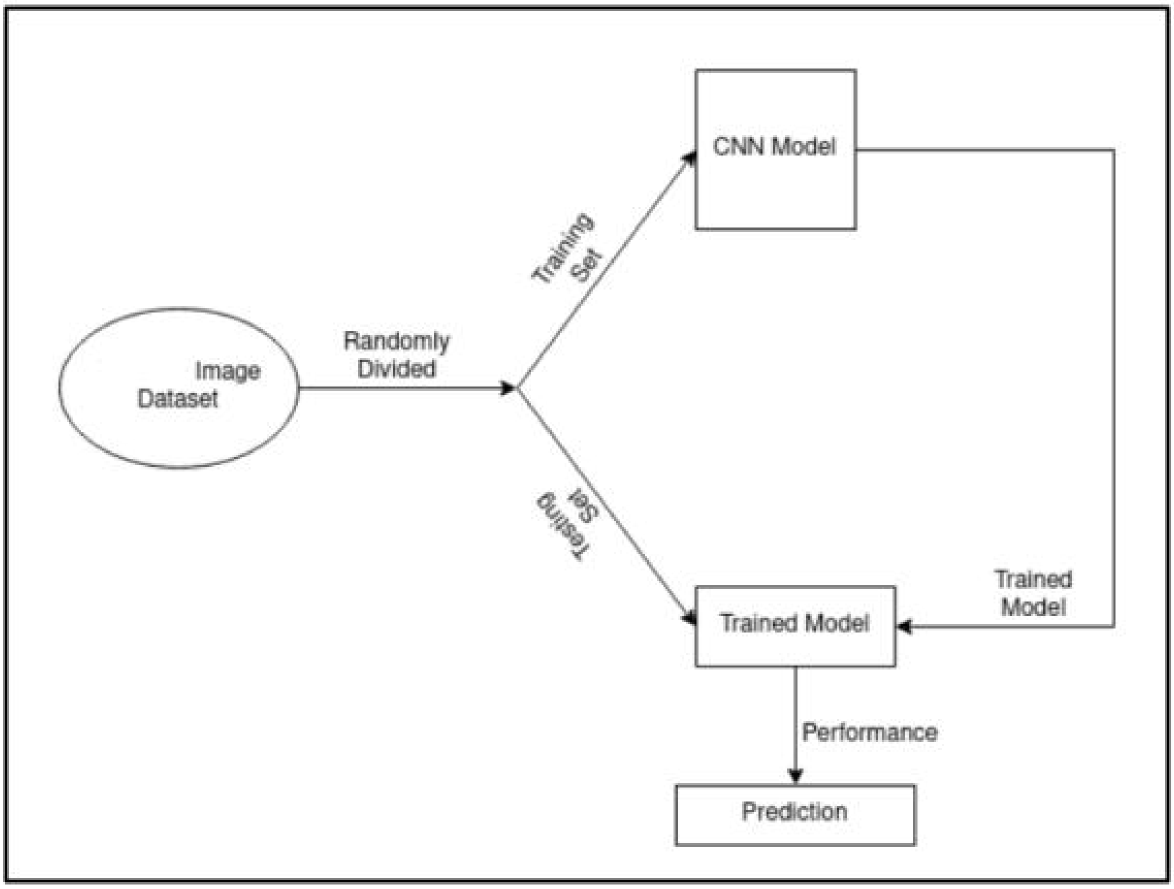

Fig. 9. Proposed Architecture

Firstly, we will divide our soybean image dataset randomly between training set and testing set. Now we will train our CNN model by changing parameters and adjusting layers in the model. Then the trained model will be used as the final model which will be used for validating and predictions will be made on that.

Chapter-5

Results Vizualization

5.1 Training And Validation Accuracy

A graph in Fig. 10 shows the accuracy is plotted for 6 epochs after training and validation processes. The blue line in the graph indicates the training accuracy while black dots line is the validation accuracy.

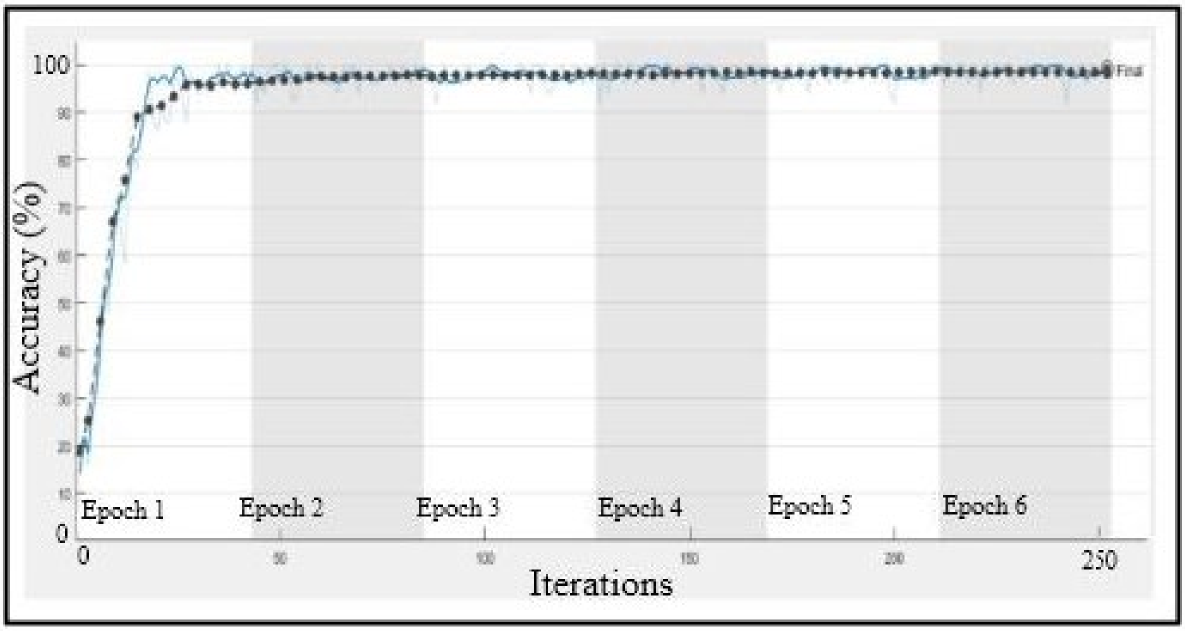

Fig. 10. Plot of training and validation accuracies with number of iterations. (Accuracy vs Iterations)

A training accuracy is the accuracy obtained after performing classification on the training dataset using the trained model. The training accuracy is generally higher than validation accuracy because the classification is applied on the same dataset where the model learned from. In this case, after performing classification on the training dataset, the training accuracy obtained is 98.97%. However, the validation accuracy will be emphasized more in this case as it measures the capability and quality of the trained model. The required training duration is 210.37 minutes. It is observed that the accuracy started to get stable after the first epoch of training. With only a single time of training on the entire dataset of training images, the model is able to learn the majority of the features and classify the images in the validation set. As shown in Fig. 10, the validation accuracy obtained is 98.47%. The validation accuracy also slowly improves as the number of epochs increases. Finally, it reaches approximately 97% after 6 repeated training on the entire set. As plotted in Fig. 11, the losses after the prediction from training and validation are also investigated. A loss is a value indicating how bad the classification is done on the input image. If the prediction of the trained model is perfect, then the value of loss is 0. Otherwise, the value of loss is greater. By observing the graph, the value of loss is getting smaller as the number of training iterations increases. This means that the model is getting better on the classification of images after more training is applied to it. In this case, the value of loss is approximately 0.1 which is a small value of loss.

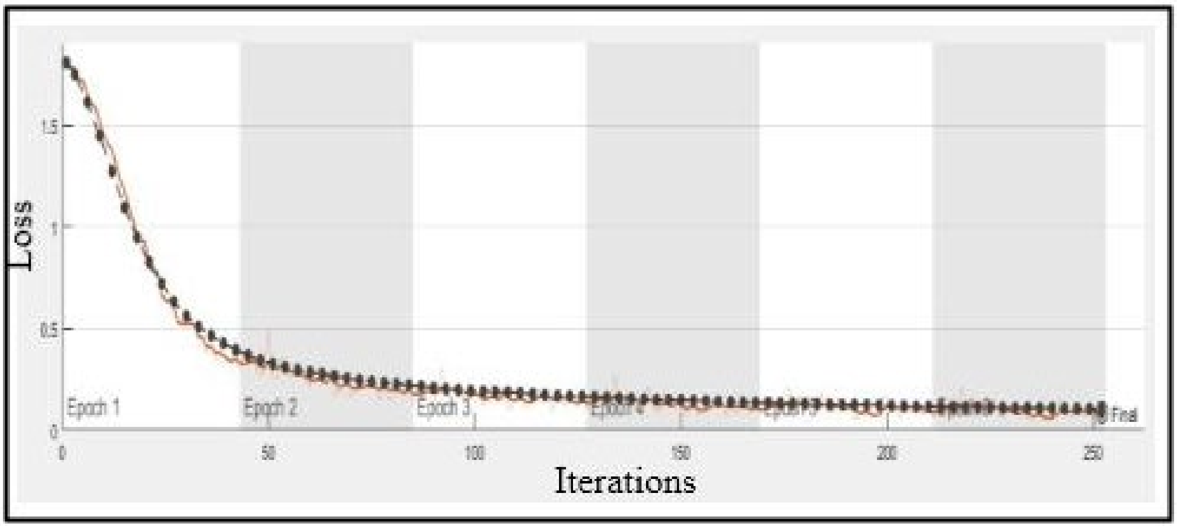

Fig. 11. Plot of training and validation losses with number of iterations. (Loss vs Iterations)

5.2 Some Input Images With Output Prediction

Fig. 12 shows the prediction result on 4 random images from the validation set. The classification results for these four images are predicted correctly with a high probability. Taking the first image as example, the model has detected the image as rotten apple with 94.2% accuracy. There is only 5.8% that this image is recognized as other categories of fruits.

The color and texture of rotten apple and oranges are not clearly distinguishable. The extracted features are similar, therefore, it is reasonable for the trained model to classify it wrongly. There are two alternate methods to solve this. The first method is to increase the amoDunt of training dataset for rotten apples and rotten oranges. This will allow the model to learn better and differentiate the features between the rotten apple and orange. However, this method depends on the availability of image dataset and also requires more time to train. The second

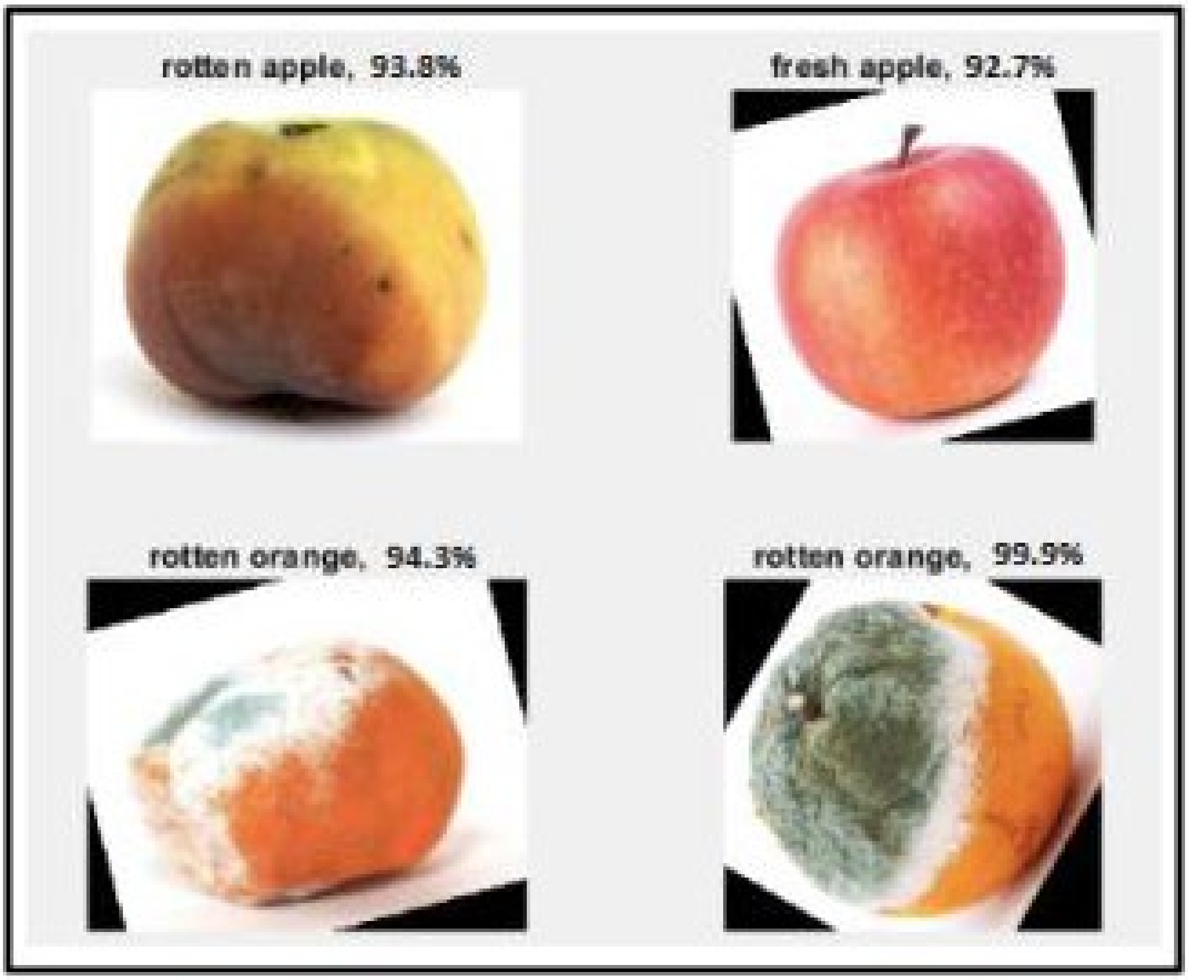

Fig. 12. Four random images from validation set with their prediction result

method is to separate the model where one model is mainly trained to classify orange while another model is to classify apple. The method can address the problem easily with minor modification in the model coding. Furthermore, there are 2 images that show the classification on the type of fruits are correct but with a wrong prediction on the condition of fruits. This error could bring an issue compared to another 7 images which only wrongly predicted the type of fruits. The reason is because it will cause the rotten fruits to pass over and sell to the customer, leading to possible customers' health issues if the system is deployed for use in the market.

Fig. 14 shows the probability of each class after classification for the selected image that has been classified wrongly as a rotten apple. In fact, the image is a fresh apple. The probability of the image projected as a fresh apple is 36.55% while rotten apple is 47.14%. The difference in the probability is 10.59%. Based on the observation on the training dataset, the majority of the images are apples at the best ripening stage which is fully red apple or green apple. In contrast, only a minority of the images are apples with partially yellow and partially red color on the apples' surface. With less amount of unripe apple images, the model could not learn well and recognize it correctly. Therefore, more images of this kind of apple should be included in the training set. From our experiment studies, the probability of rotten green apples being recognized wrongly as fresh apples is also high. This is also due to the small amount of rotten green apple images available in the training dataset. Similar solution is to increase the number of rotten green apple images in the training dataset that can be trained to reduce the probability of wrong prediction. By doing this, the model could learn the features of rotten green apples and to recognize them correctly.

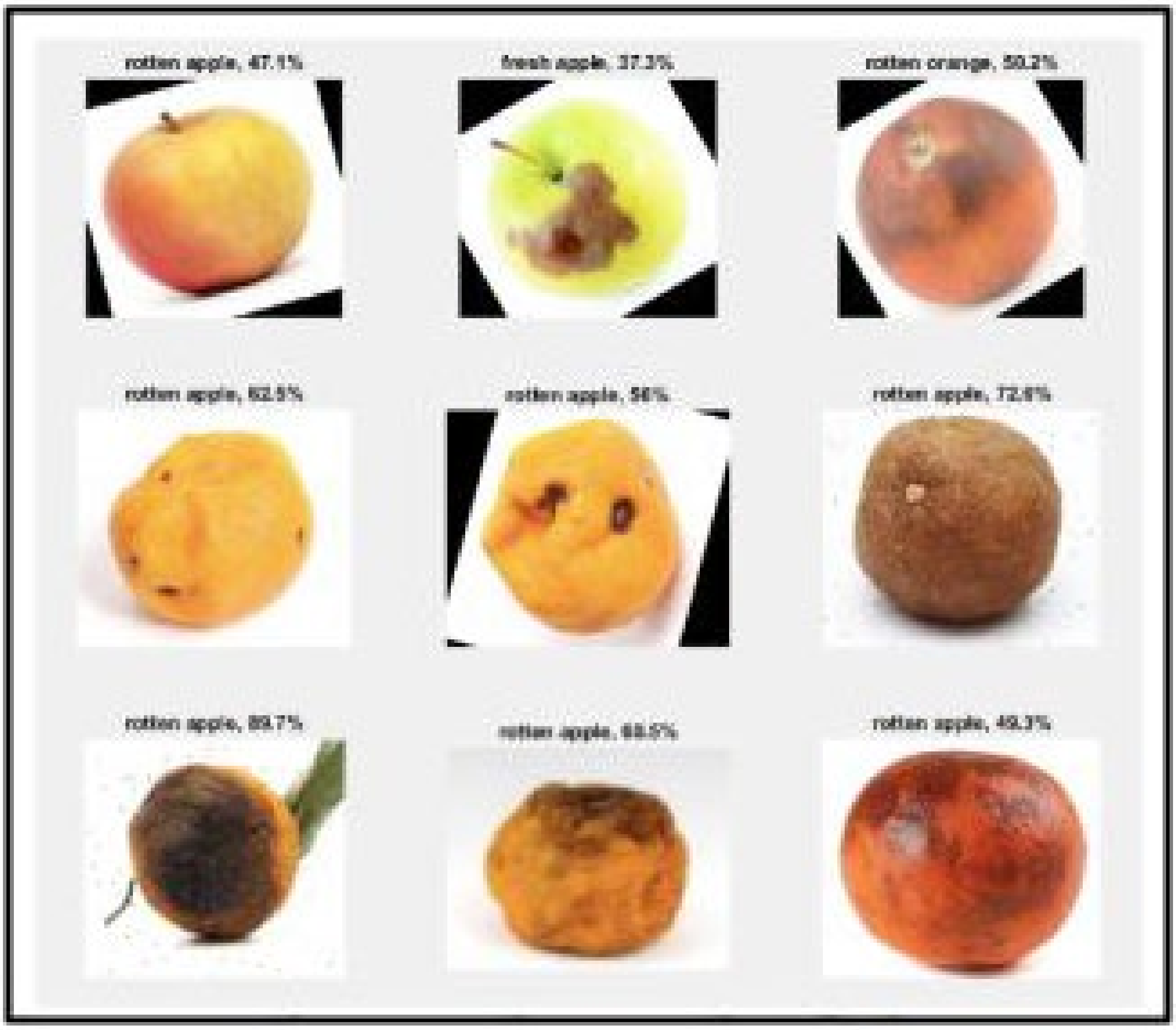

Fig. 13. Images that are wrongly predicted

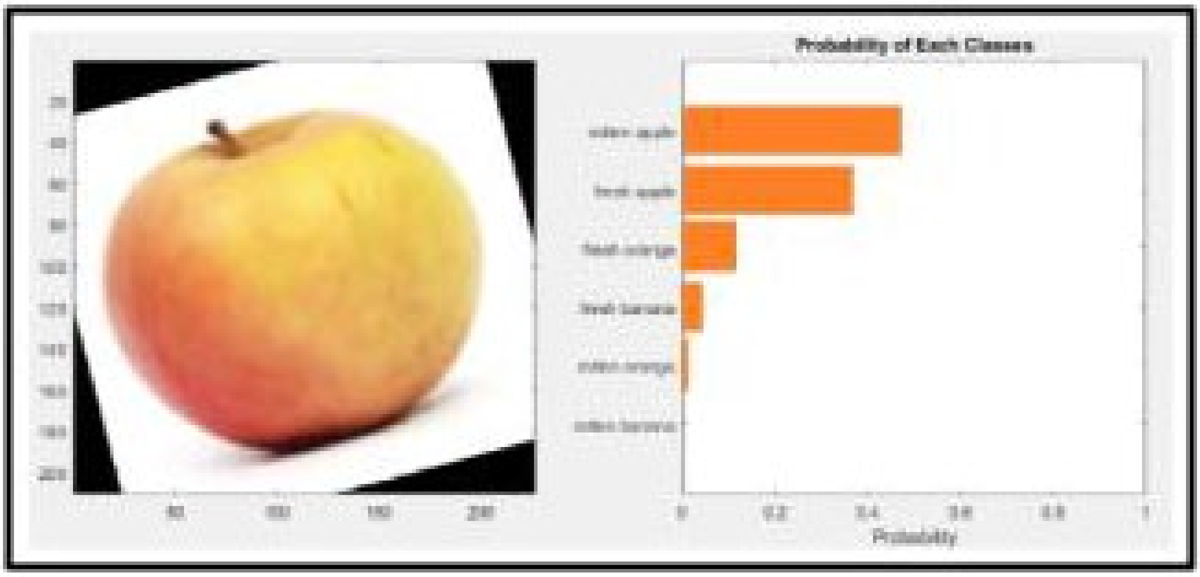

Fig. 14. Probability of each class after classification

5.3 Results Comparision On Different Model Parameters

the training option parameters for detecting rotten apple, orange and banana using 350 training datasets for each category are determined. They are listed as shown in Table III. Further experiments have also been conducted to investigate the effect of varying mini-batch size and the number of epochs to the validation accuracy and training duration when training with Inception v3. The mini-batch size of 10, 50 and 100 are for investigation. Additionally, the number of epochs to be deployed in the experiment are 3 and 6 epochs. The reason for not using more number of epochs in this case is because if the number of epochs used to train a neural network is more than necessary, the training model will learn the pattern that are specific to a great extent with longer training time than 210.37 minutes using 6 number of epochs in this experiment. Increasing the number of epochs will cause overfitting of the training data. Therefore, 6 number of epochs is considered as an optimal value to obtain little overfitting to a training data with good accuracy and acceptable training time. On the other hand, a smaller mini- batch size tends to give better validation accuracy as the frequency of updating the weights of the model is higher. The model is able to learn faster and better when the weights are updated more frequently. The result will be discussed in the next section to demonstrate the validation of the theoretical studies.

Based on Table IV, it shows that by setting the mini- batch size as 10, the validation accuracy obtained is the highest among other cases which is 98.47%. The results obtained from the experiment have agreed well with the theory. With the mini-batch size as 10, the training duration is much longer compared to the training duration for 50 and 100 mini-batch size. This is because the process of updating the weights of the model consumes time while the weights are updated every iteration. For case 1, the number of iterations is 252, which indicates the weights of the model are updated for 252 times. Table V investigates the relationship between the number of epochs and the validation accuracy with different numbers of iterations. It shows that the validation accuracy for 6 epochs is 98.47%, which is higher than using 3 epochs for validation. Based on theoretical study, a large number of epochs results in higher accuracy as the larger number of epochs indicates more number times that the learning algorithm works through the entire training dataset. On the other hand, with a large number of epochs, the training duration is longer due to the fact that the iteration process will take more time to update the internal model parameters through forward pass and backward pass of all training datasets. In the results, the training duration recorded is 23.7 minutes for 3 epochs, while 44.7 minutes for 6 epochs.

TABLE III. Parameters of training option

Mini-batch Size	Number of Epochs	Initial Learning Rate	Validation Frequency
50	6	0.0001	3

Table IV. Comparison of results in validation accuracy for 10, 50 and 100 mini-batch sizes

Sr. No.	Mini-batch Size	Number of Epochs	Number of Iteration	Validation Accuracy	Training Time
1	10	6	252	98.47%	44.72
2	50	6	48	96.88%	12.59
3	100	6	24	94.68%	9.51

Table V. Comparison of results in validation accuracy for two different number of epochs

Sr. No.	Mini-batch Size	Number of Epochs	Number of Iteration	Validation Accuracy	Training Time
1	10	3	126	95.93%	23.68
2	10	6	252	96.88%	44.72

5.4 Application Snapshorts With Different Test Cases

Test Case 1 ::

System Input : Image of Fresh Apple

Expected Output : Fresh

System Output : Fresh

Verdict : Passed

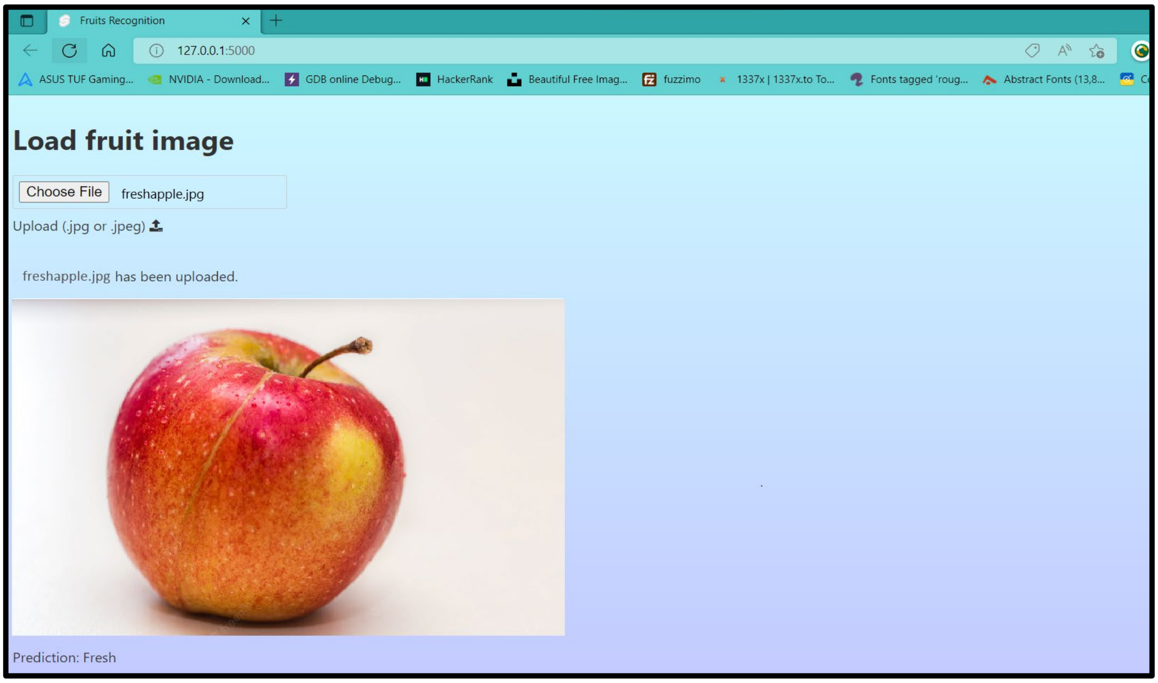

Fig. 15. Application Screenshot For Test Case 1

Test Case 2 ::

System Input : Image of Rotten Apple

Expected Output : Rotten

System Output : Rotten

Verdict : Passed

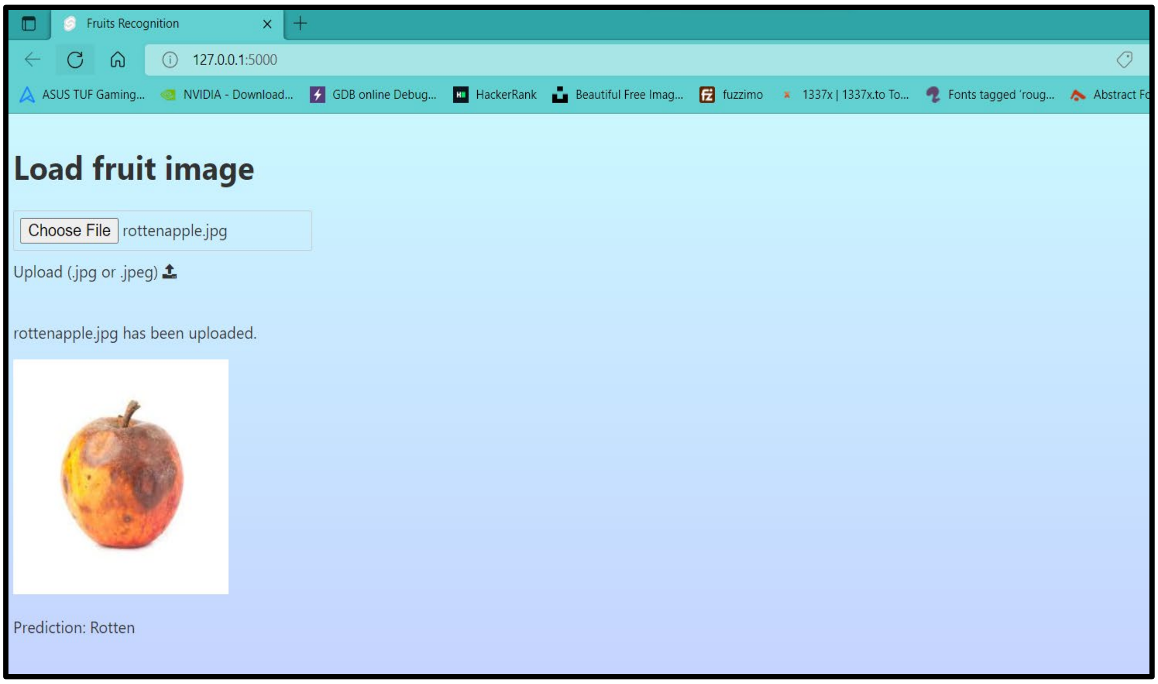

Fig. 16. Application Screenshot For Test Case 2

Test Case 3 ::

System Input : Image of Fresh Banana

Expected Output : Fresh

System Output : Fresh

Verdict : Passed

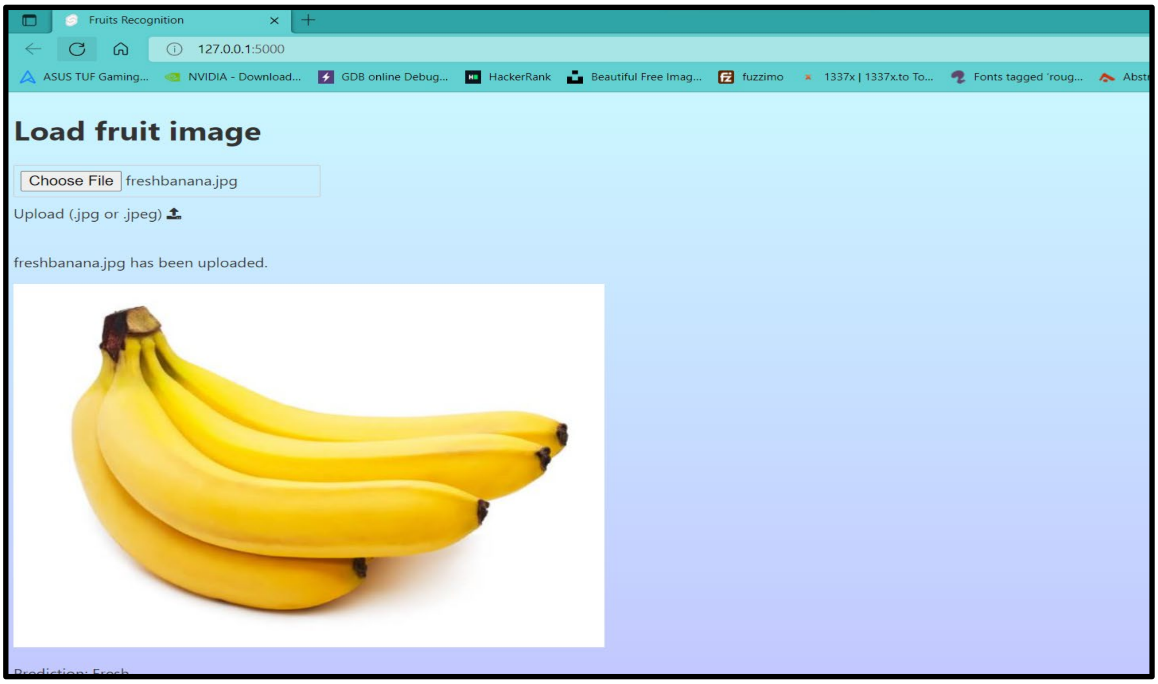

Fig. 17. Application Screenshot For Test Case 3

Test Case 4 ::

System Input : Image of Rotten Banana

Expected Output : Rotten

System Output : Rotten

Verdict : Passed

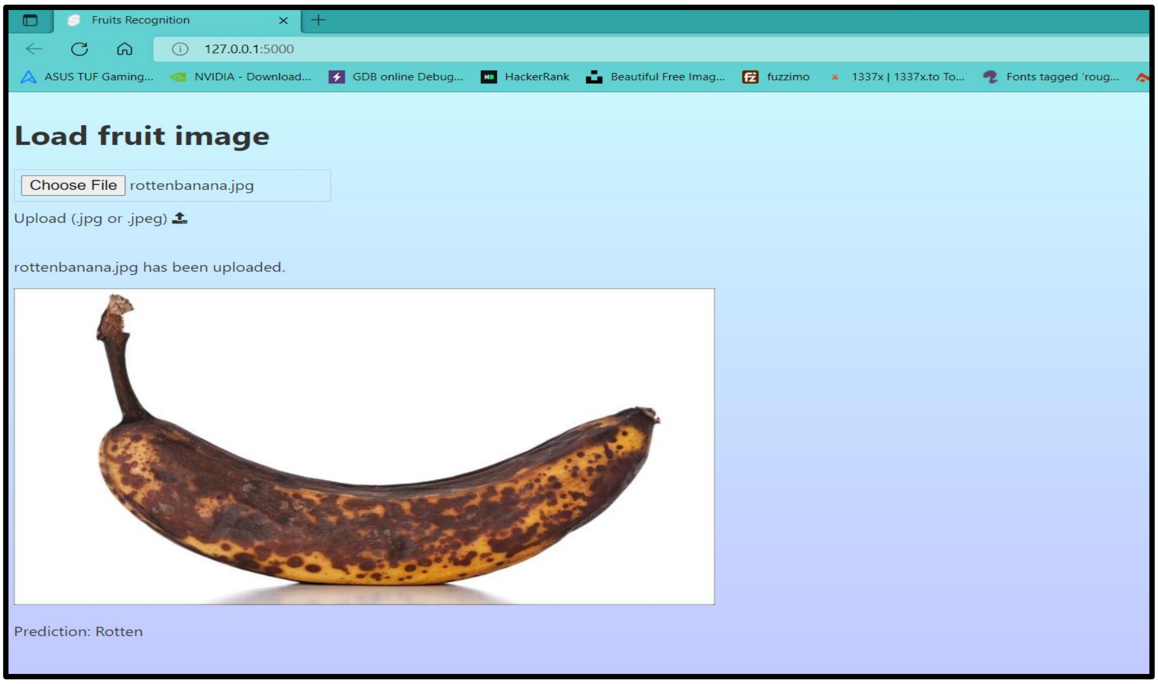

Fig. 18. Application Screenshot For Test Case 4

Test Case 5 ::

System Input : Image of Fresh Orange

Expected Output : Fresh

System Output : Fresh

Verdict : Passed

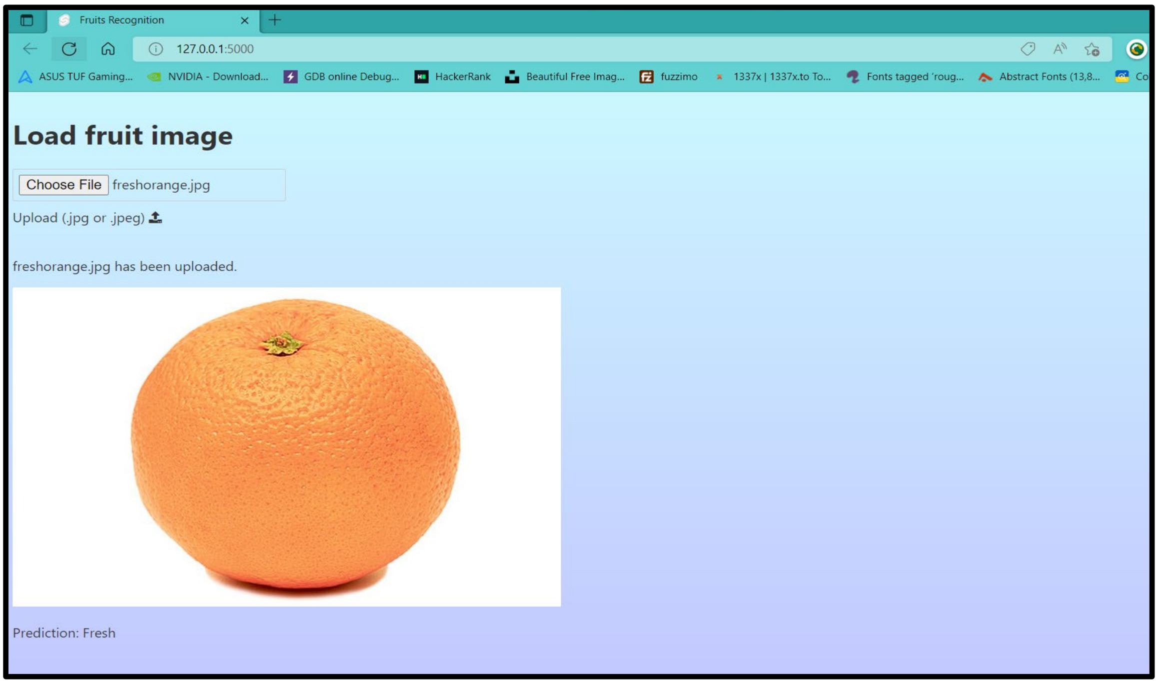

Fig. 19. Application Screenshot For Test Case 5

Test Case 6 ::

System Input : Image of Rotten Orange

Expected Output : Rotten

System Output : Rotten

Verdict : Passed

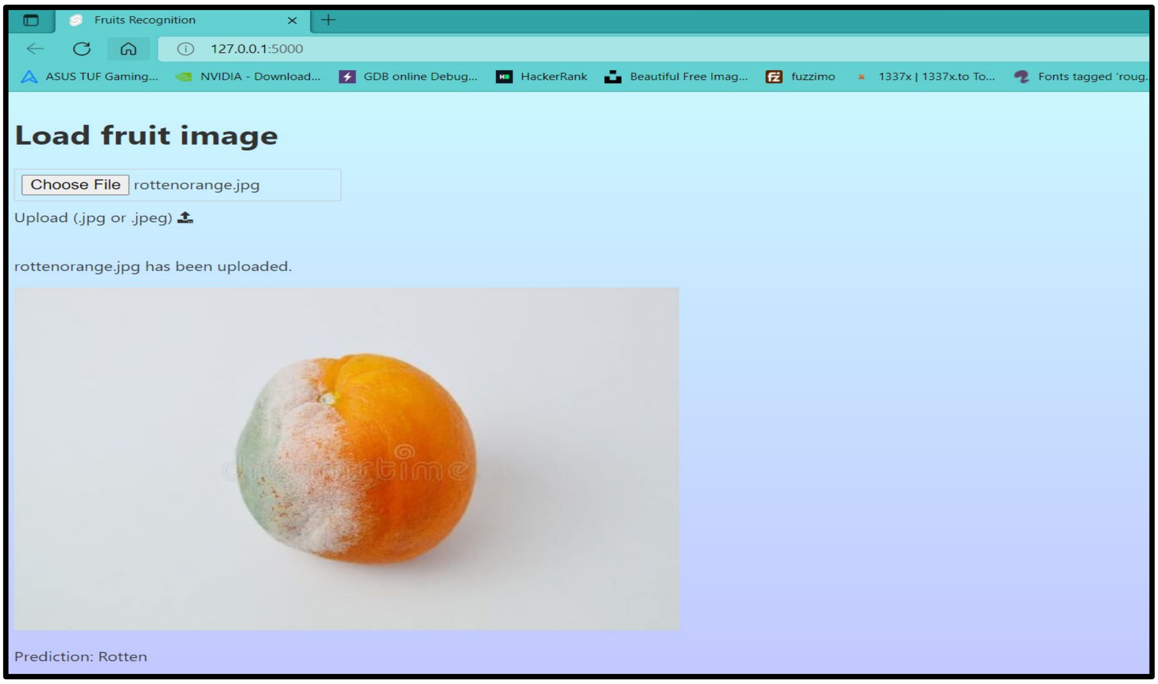

Fig. 20. Application Screenshot For Test Case 6

Chapter-6
Conclusion

6.1 Discussion

As a conclusion, the proposed method applied is to use transfer learning to detect the rotten fruits. A pretrained network, Inception v3 is used to retrain by changing the final layer to the desired output. The total training duration required to train the model with 3000 input images set is 210.37 minutes. The training duration is long, but the model will take even longer for training if it is developed from scratch instead of using a pretrained network model. The duration of training can be shorter if higher performance of computers is used. The trained model has its own good flexibility because the trained model can be saved as a network to retrain it. When a new set of input images is desirable to be added into the training dataset to improve the accuracy, the training process with the use of previous original dataset does not need to repeat. Instead, just load the trained model and train it using the new set of images data. In additional work, a new category of fruit such as strawberry can also be included for classification using this model. Firstly, load the pretrained network which has already contained the features of six categories of the original fruit images. Then, change the final layer to indicate there are seven output classes and then follow by training the network using the dataset with strawberry images. This shows the flexibility of the transfer learning technique.

Although a good accuracy of 98.47% is obtained in detecting the rotten apple, orange and banana. There is a limitation that affects the performance of classification using this proposed method, which is the requirement specified on the image dataset. For training with the proposed model in this paper, it is important to make sure that the background of images is a monotonous color. This is because if the background of a fruit image is non-monotonous and is contaminated with noises, it is required to be processed, filtered and segmented in order to extract the features of a fruit in the image properly. Else, it will act as an unwanted feature which will be learned by the model during the training stage. This could affect the result of classification. Based on the result of classification, majority of the wrongly predicted images are under the category of rotten orange and apple. This showed the difficulty of the trained model to classify between these two categories as the appearance of rotten orange and apple on the outer surface has a similar pattern and color. These similar features being obtained by the model for rotten apple and rotten orange led to wrong classification.

6.2 Final Remarks

An intelligent rotten fruit detection system has successfully developed. The CNN algorithm was studied appropriately in order to detect the rotten fruits. The transfer learning technique was used where the model was retrained from a pre-trained CNN network Inception v3. The trained model was then used to perform classification on the rotten and fresh fruits. A total of 2100 fruit images with monotonous backgrounds were trained. After comparing the accuracy from the experiment, the input dataset applied for both training and validation stage does not require segmentation. The training option parameter is determined as 6 number of epochs and 30 mini-batch size. With these settings, the required duration for the training stage is 210.37 minutes. Six categories of classification were included in this paper including rotten and fresh fruits for apple, banana and orange respectively. The validation accuracy obtained is 98.47%. This indicates that there are only 9 images that were classified wrongly from out of 900 images in the validation dataset. In terms of accuracy, classifying a fresh or rotten banana can attain a better result than orange and apple. This is because all the wrongly predicted images are in the category of apple and orange. Evaluating in terms of the speed of the classification process, it requires approximately 2 seconds to complete the classification on 10 fruit images. After performing classification on 60 fruit images with non-monotonous background, it has been observed that the accuracy has dropped from 98.47% to 85%.

6.3 Future Scope

In future work, it is expected that the probability of correct prediction on green apples can be improved with more apple images having different levels of ripeness to be included into the training dataset. This increased amount of fresh and rotten apple images can reduce the possibility of obtaining wrong classification. As the difficulty of classifying rotten apples and rotten orange was discussed. It is recommended to separate the trained model into two different models, one model is to focus on classifying apples while another is to classify solely on orange. With this method, the accuracy can expect to be higher in classifying the apple and orange.

REFERENCES

[1] Isikdogan, “Transfer Learning”, 2018, http://www.isikdogan.com/blog/transfer-learning.html

[2] L. B. Aydilek, "Approximate estimation of the nutritions of consumed food by deep learning," in 2017 International Conference on Computer Science and Engineering (UBMK), 2017, 160-164

[3] K. Dharavath, G. Amarnath, F. A. Talukdar and R. H. Laskar, "Impact of image preprocessing on face recognition: A comparative analysis," in 2014 International Conference on Communication and Signal Processing, 2014, pp. 631-635.

[4] Prabhu, "Medium - Understanding of Convolutional Neural Network (CNN) — Deep Learning," 2018. [Online].

[5] Wang, L., Li, A., Tian, X. (2013). Detection of fruit skin defects using a machine vision system. In 2013 Sixth International Conference on Business Intelligence and Financial Engineering, pp. 44- 48. https://doi.org/10.1109/bife.2013.11

[6] D. Karakaya, O. Ulucan and M. Turkan, "A Comparative Analysis on Fruit Freshness Classification," in 2019 Innovations in Intelligent Systems and Applications Conference (ASYU), 2019, pp. 1-4.

[7] A. Wajid, N. K. Singh, P. Junjun and M. A. Mughal, "Recognition of ripe, unripe and scaled condition of orange citrus based on decision tree classification," in 2018 International Conference on Computing, Mathematics and Engineering Technologies (iCoMET), 2018, pp. 1-4.

[8] M. Jhuria, A. Kumar and R. Borse, "Image processing for smart farming: Detection of disease and fruit grading," in 2013 IEEE Second International Conference on Image Information Processing (ICIIP-2013), 2013, pp. 521-526.

[9] Stevo Bozinovski, “Transfer Learning in Neural Networks”. https://www.informatica.si/index.php/informatica/article/view/2828/1433

[10] K. R. Sriram, "Kaggle - Fruits fresh and rotten for classification," 2018. [Online]. Available: https://www.kaggle.com/sriramr/fruits fresh-and-rotten-for-classification.

[11] S. Trambadia and H. Mayatra, "Food detection on plate based on the HSV color model," in 2016 Online International Conference on Green Engineering and Technologies (IC-GET), 2016, pp. 1-6

[12] I. Z. Mukti and D. Biswas, "Transfer Learning Based Plant Diseases Detection Using ResNet50," in 2019 4th International Conference on Electrical Information and Communication Technology (EICT), 2019.

[13] "Inception v3 Model Architecture" [Online] https://cloud.google.com/tpu/docs/inception-v3-advanced

[14] Azizah, L.M.R., Umayah, S.F., Riyadi, S., Damarjati, C., Utama, N.A. (2017). Deep learning implementation using convolutional neural networks in mangosteen surface defect detection. In 2017 7th IEEE International Conference on Control System, Computing and Engineering (ICCSCE), pp. 242-246. https://doi.org/10.1109/iccsce.2017.8284412