A DEEP LEARNING MODEL FOR CROP CLASSIFICATION

NORADILA HANIS BINTI HAIRUDIN

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

A DEEP LEARNING MODEL FOR CROP CLASSIFICATION

NORADILA HANIS BINTI HAIRUDIN

This report is submitted in partial fulfilment of the requirements for the degree of Bachelor of Electronic Engineering with Honors

Faculty of Electronics and Computer Engineering
Universiti Teknikal Malaysia Melaka

2022

DECLARATION

I declare that this report entitled "A DEEP LEARNING MODEL FOR CROP CLASSIFICATION" is the result of my own work except for quotes as cited in the references.

Signature : …………………………………

Author : NORADILA HANIS BINTI HAIRUDIN

Date : 11 January 2022

APPROVAL

I hereby declare that I have read this thesis and, in my opinion, this thesis is sufficient in terms of scope and quality for the award of Bachelor of Electronic Engineering with Honours.

Signature : ....

Supervisor Name : DR. FAKRULRADZI BIN IDRIS

Date : 11 January 2022

DEDICATION

This project report is dedicated to my wonderful family and supervisor. They have always been there for me. They aided me in completing my job successfully. Moreover, I will devote this project report to a certain topic area. The project report is meant to be a resource for future research and analysis.

ABSTRACT

Given the importance of food status in modern civilization and its economic contribution, it becomes more critical to improve fruit freshness, but manual operation is time demanding. On the one hand, each crop falls into a number of difficult-to-categorize categories. The purpose of this project is to determine how successfully an artificial intelligence can classify them. The eggplant was chosen as the crop for this research. The algorithm is designed to be capable of evaluating fruit quality based on metrics such as size, colour, shape, and intensity, however colour and size remain the most critical factors in fruit grading and sorting. Thus, this project will develop a model that capable of automatically determining the freshness of eggplants, which will save time, labour, and provide a higher level of accuracy than manual sorting. The primary objective is to classify the eggplants produced in terms of their quality. When three grades of eggplants are input into the deep learning system. It is self-adjusting in order to classify (grade) eggplants depending on sample sets. When an eggplant has characteristics of one of the three categories, it is categorised and graded appropriately. This technique is quick and dependable; more importantly, it produces consistent results. A total of 148 eggplant datasets were collected in order to obtain the results.

ABSTRAK

Memandangkan kepentingan status makanan di masa kini dan sumbangan ekonominya, menjadi lebih penting untuk meningkatkan kesegaran buah, tetapi operasi manual adalah sangat melecehkan. Setiap tanaman jatuh ke dalam beberapa kategori yang sukar dikategorikan. Terung dipilih sebagai tanaman untuk penelitian ini. Algoritma ini direka untuk menilai kualiti buah berdasarkan metrik seperti saiz, warna, bentuk, dan intensiti, namun warna dan saiz tetap menjadi faktor yang paling utama dalam pengasingan buah. Oleh itu, projek ini akan membina satu model yang mampu menentukan kesegaran terung secara automatik, yang akan menjimatkan masa, buruh, dan memberikan tahap ketepatan yang lebih tinggi daripada pengasingan manual. Objektif utama adalah untuk mengklasifikasikan terung yang dihasilkan dari segi kualiti mereka. Apabila tiga gred terung dimasukkan ke dalam sistem pembelajaran yang mendalam. Ia menyesuaikan diri untuk mengklasifikasikan (gred) terung bergantung kepada set sampel. Apabila terung mempunyai ciri-ciri salah satu daripada tiga kategori, ia dikategorikan dan digredkan dengan sewajarnya. Teknik ini cepat dan boleh dipercayai; yang lebih penting, ia menghasilkan keputusan yang konsisten. Sebanyak 148 set data terung telah dikumpulkan untuk mendapatkan keputusan.

ACKNOWLEDGEMENTS

In the name of Allah S.W.T., the Most Gracious and Merciful. First and foremost, I am extremely grateful to Allah S.W.T for His gracious blessings and guidance, as well as for providing me with the strength to successfully finish my final year project and report. Alhamdulillah and thanks to Allah S.W.T as this report can be done within the given time.

First and foremost, I would want to convey my heartfelt gratitude and appreciation to my supervisor, Dr. Fakrulradzi Bin Idris. for his consistent support and assistance throughout this project. His understanding of issues and experience in this field of study aided me in gaining a better understanding of the study, conducting the research appropriately, and presenting the findings clearly in the report. Without his direction and emotional support during the year, I'm not sure I'd be able to complete this study.

Finally, I would like to express my gratitude to my parents, and a friend for their assistance, compassion, patience, encouragement, and dedicated direction during the writing of the report. I am grateful to all of you.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS AND ABBREVIATIONS

CNN	:	Convolutional Neural Network
RNN	:	Recurrent Neural Network
MPL	:	Multi-Layer Perceptron
NN	:	Neural Networks
IOT	:	internet of things

CHAPTER 1

INTRODUCTION

1.1 Project introduction

Crop failure has a significant economic impact. Given the importance of food status in today's culture and its economic contribution, improving fruit freshness becomes increasingly vital, yet manual operation is time intensive. Automation grading by a computerized technique is seen to be the answer to this dilemma. We picked the job of classifying eggplant at the start of this research for a variety of reasons. On the one hand, eggplant is classified that are difficult to identify. As a result, we want to examine how well artificial intelligence can classify them. Our primary goal is to classify the grade of eggplants produced. When the deep learning system is given three different grades of eggplants. Based on the sample sets, it is self-adjusted to categorize (grade) the eggplant. When an eggplant resembles one of the three classes, it is categorized and graded accordingly. This approach is quick and dependable; more importantly, it produces consistent results.

1.1.1 Eggplants export statistic for Malaysia

Malaysia is Australia's major eggplant export market, accounting for around 69% of total export value in 2015-16. Malaysia also obtained the highest price per kilograms for eggplants in 2015-16, at $8.20, or RM 24.71.

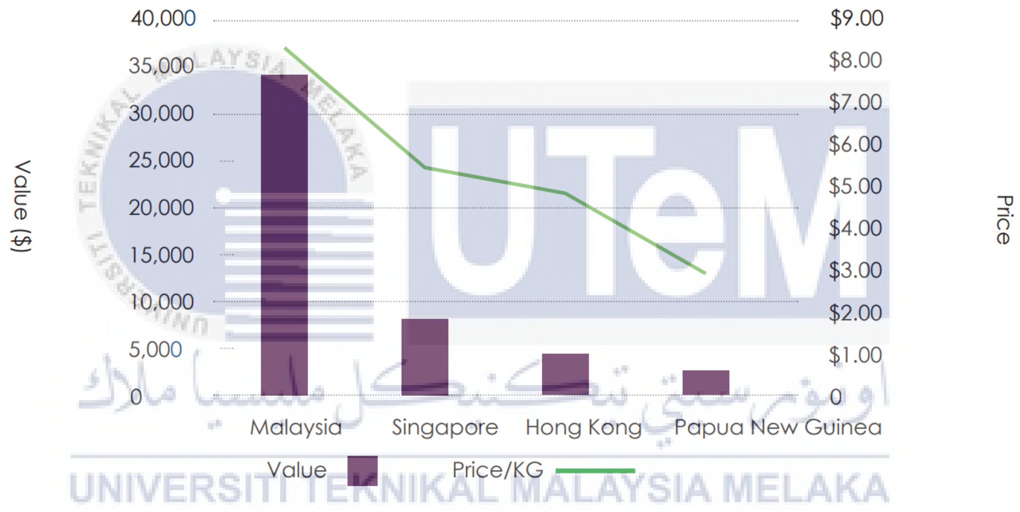


Figure 1 : Export Markets 2015-16

Not only that, the Maldives, Germany, France, Switzerland, and the Netherlands were among the best performing markets in 2019 for Malaysian aubergines (eggplants). Malaysian exports of aubergines (eggplants) are classified as fresh or chilled aubergines "eggplants".

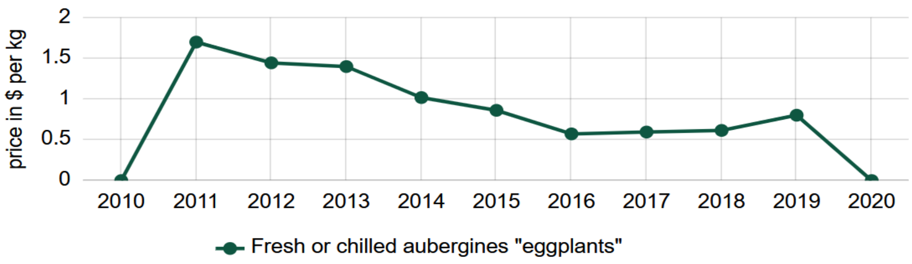


Figure 2 : Export Price of Malaysia Eggplants

1.2 Problem statement

Brinjal, commonly known as eggplant, is a popular tropical vegetable that is grown. A deep learning-based brinjal crop yield prediction system is a potential option for measuring the success of a fertigation system. It also offers an automated method of determining the quality of Brinjal produced. Manual grading and separating any fruit are a challenge. Because it is a labor-intensive and time-consuming procedure, it causes farmers to incur expenditures by delaying post-harvest operations. Aside from that, human examination causes additional issues in maintaining grade consistency and uniformity while compiling. Manual techniques, on the other hand, are laborious and error-free.

1.3 Objective

i. To develop a deep learning model for eggplant classification.

ii. To increase the accuracy and perfection of the fruit freshness detection with the help of size, shape, and color-based method with the union of Convolutional Neural Network (CNN)

iii. To validate the performance of the classification technique of the Deep Learning model in terms of accuracy.

1.4 Scope of work

TensorFlow and Keras are used as libraries in this project. A convolutional neural network (CNN) is a type of deep neural network that is commonly used to evaluate visual images in deep learning. Python is the coding language used. Linux (Ubuntu) is the operating system.

Table 1 : Scope of Work

Item description	Description
Library	TensorFlow and Keras
Deep Learning	CNN
Coding Language	Python
Operating System	Linux (Ubuntu)

CHAPTER 2

BACKGROUND STUDY

2.1 Artificial Intelligence

It is the capability of a computer programmed to learn and think that is referred to as Artificial Intelligence. Artificial intelligence (AI) is a database technology that work together to enable machines to sense, comprehend, act, and learn at levels of intelligence comparable to those of humans. AI is defined as the ability of machines to sense, comprehend, act, and learn at levels of intelligence comparable to those of humans. Expert systems, natural language processing, voice recognition, and machine vision are some of the specific applications of artificial intelligence. Artificial intelligence algorithms are programmed that are meant to make judgments based on data that is frequently available in real time. They are different from passive machines, which are limited to mechanical or programmable outputs.

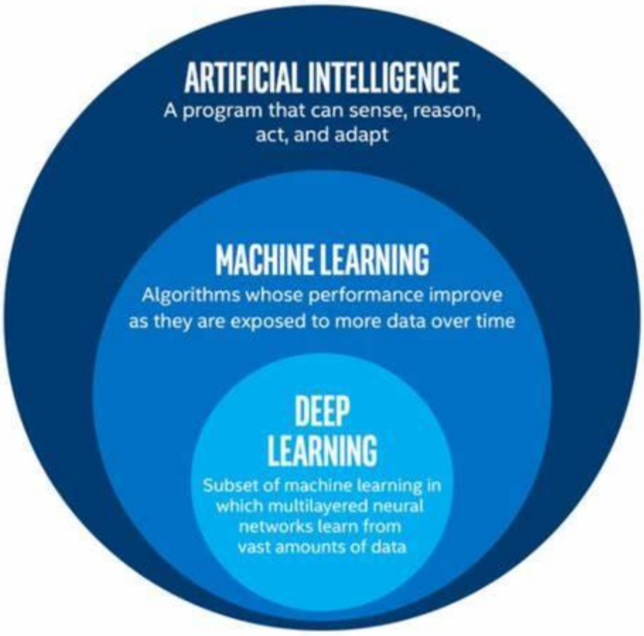


Figure 3 : AI, Machine learning and Deep learning

2.2 Deep Learning

Deep learning is a subset of machine learning, and it is becoming more popular. Deep learning is seen as the next step in the development of machine learning. The term "deep neural network" refers to a neural network that has more than one layer, which is more than one in number. And this is what the term "deep learning" refers to in this context. A deep learning model is a machine learning system that is developed using a deep neural network as the learning engine. Because of the programming of the neural network, robots are capable of making precise judgments without the assistance of humans.

In order for deep learning to learn from its input, each algorithm in the hierarchical order performs a convolution, and then utilizes what it has learned to produce a prediction approach as an output. Iterations are repeated until the output has attained an acceptable degree of accuracy. Iteration is the number of batches or steps through partitioned packets of training data that are required to complete one epoch of training. The term deep refers to the large number of processing layers that data must travel through before being used.

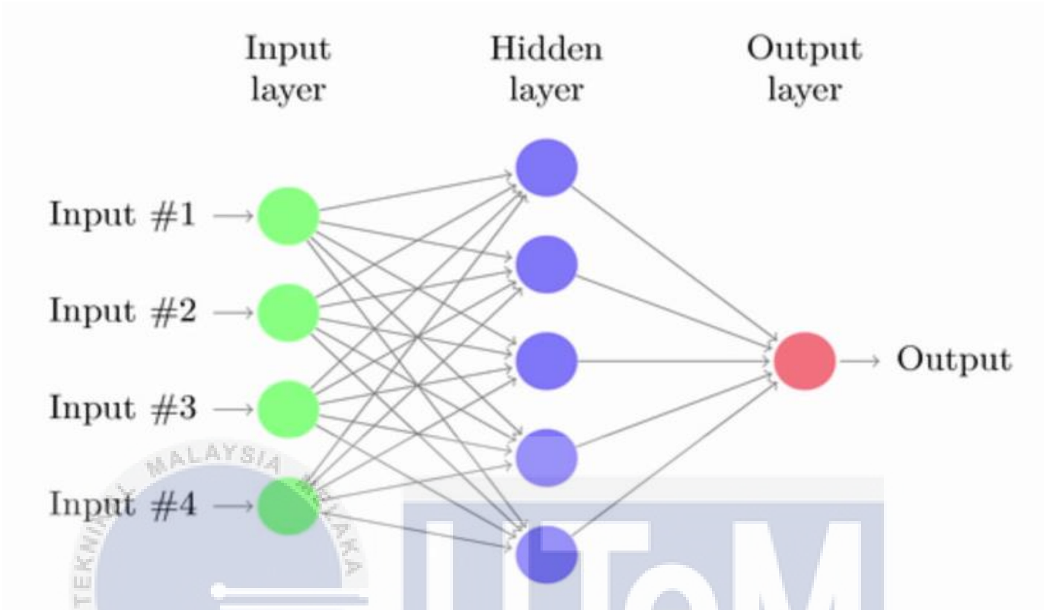


Figure 4 : Deep Learning Model

2.3 Machine Learning

In artificial intelligence (AI) and computer science, machine learning is a subfield that focuses on the use of data and algorithms to mimic the way that people learn, with the accuracy of the imitating system continuously increasing over time. A prediction or classification is made with the use of machine learning algorithms, which are often utilized in this context. If there are known examples, an error function may be used to do a comparison in order to determine the correctness of the model. I n the field of machine learning whenever the model can suit specific the data points in the training set, the weights are updated to lessen the gap between the known example and the estimated value from the model. After completing this evaluation and optimization process, the algorithm will update weights on its own, repeating the process until a certain level of accuracy has been achieved.

A supervised, semi-supervised, or unsupervised environment may be used to facilitate learning and development. In machine learning, supervised machine learning is characterised by the use of labelled datasets to train algorithms that are accurate in classifying data or in predicting outcomes. As new data is introduced into the model, the weights of the model are adjusted until the model is well suited to the new data. It is possible to find hidden patterns or data groupings using these algorithms without the requirement for human participation. Unsupervised learning techniques include neural networks, k-means clustering, probabilistic clustering approaches, and other methods of classification. Learning under semi-supervision is an excellent middle ground between supervised and unsupervised learning situations. A semi-supervised learning method may be used to alleviate the issue of not having enough labelled data (or not having the financial means to label enough data) to train a supervised learning algorithm.

2.4 Convolutional Neural Network

A Convolutional Neural Network (CNN) is a kind of multilayer neural network that consists of one or more convolutional layers (sometimes with a subsampling step) followed by one or more fully connected layers, similar to a regular multilayer neural network. The architecture of a CNN is created in such a way that it can take advantage of the 2D structure of the picture input (or other 2D input such as a speech signal). An effective convolutional neural network has millions of parameters and many hidden layers and is a behemoth of computing power. As a matter of fact, the greater the number of hidden layers, the more effective the network." Some of the most well-known networks are AlexNet, VGG, Inception, and ResNet.

2.5 Literature review

Deep learning is used in a variety of activities, including movie recommendation systems, spam detection, and computer vision. Though there is continuous debate about deep learning as a black box and the difficulty of training, it has enormous potential in a wide range of disciplines. Deep learning approaches, for example, have been widely employed in medical imaging research in the form of several effective classifier and grouping algorithms [1]. Convolutional Neural Network is a deep neural network class that is most often used to evaluate visual pictures in deep learning. In agriculture, for example, deep learning is used to categorize the maturity of oil palm fruit[2], which is subsequently used to build palm fruit picking machines that can harvest oil palm fruit based on the stage of maturity[3]. Other research has shown that deep convolutional neural networks can execute leaf counting tasks[4]

Table 2 : Literature Review

NO	TITLE	DESCRIPTION
1	Tomato crop disease classification using pre-trained deep learning algorithm	In this work, the author uses a rapid, reliable, non-destructive approach that will help farmers to fight the problem of early disease detection. Images of tomato leaves (6 illnesses and a healthy class) taken from the Plant Village dataset are used as input to two deep learning-based architectures, Alex Net and VGG16 net, in this work.[5]
2	Crop Disease Detection using Deep Convolutional Neural Networks	In this article, the method works when the farmer takes a picture of a crop leaf that he has sowed in his field. After clicking, the image is submitted to the server, where it is analyzed, and the image's characteristics are retrieved. Based on such attributes, the image is classified using Neural Networks, and the result, which includes the illness name, is shown on the phone's screen. Based on the disease name, the system presents the appropriate list of Fertilizers and Organic. The suggested method will enhance crop

		efficiency by proposing suitable organic use. Drawback: VGG Net is used in this work. It takes a long time to train, and there are a lot of network architectural weights in terms of disk/bandwidth.
3	Bitter Melon Crop Yield Prediction using Machine Learning Algorithm	The purpose of this study work was to learn more about the crop bearing potential of bitter melon or bitter gourd. Images of bitter melon leaves were collected from Ampalaya fields and utilized as the primary data in the study. The description of the leaves characterized them as good or poor. The Convolutional Neural Network was utilized in the research as a Machine Learning Algorithm. Data training was accomplished by using the capabilities of Keras, Tensor Flow, and Python.[6] Drawback: Their MLP models are built using the NN tools in MATLAB. And MATLAB is mostly running in a local environment.

4	An Efficient Crop Identification Using Deep Learning	In this paper, the experiment was carried out by feeding current weather conditions such as temperature sensor values, humidity sensor values, rain sensor values, and moisture sensor values into machine learning algorithms such as random forest, decision tree, logistic regression, support vector machine, multilayer perceptron, and RNN. Then, using above mentioned algorithms, a suitable crop is found. Thus, if the farmer wishes to plant another crop, the system will be able to advise whether or not the desired crop can be cultivated.[7] Drawback: Machine learning algorithms have less accuracy when compared to deep learning.
5	Rice plant disease classification using transfer learning of deep convolution	Transfer learning of deep CNN was investigated for the first time in this work to classify rice plant diseases. Furthermore, the trials were carried out by dividing the entire dataset into different training-testing ratios. The suggested model can diagnose rice illnesses with a classification accuracy of

	neural network	91.37 percent with a training-testing partition of 80 % - 20 %.[8]
6	Machine Vision based Fruit Classification and Grading	The basic process flow of fruit categorization and grading is reviewed in this study. SURF, HOG, and LBP features are used to demonstrate feature extraction algorithms for color, size, shape, and texture.[9]
7	Image-based modeling for oil palm fruit maturity prediction	To classify the ripeness of oil palm fruit, deep learning is used. Experiments were carried out in this work to determine and model an equation between the hue optical characteristics of oil palm fruits at different stages of development after considering the influencing outdoor environment intensity in oil palm farms. The maturity phases were validated by measuring the content of mesocarp oil. This investigation was conducted on chosen immature fruits and was observed at day intervals until one loose fruit was discovered, indicating developed FFB.[2]

8	Fruit freshness detection using CNN approach.	In this paper, they examine a safe and cost-effective method for detecting fruit freshness based on size, shape, and color. Because fruits are highly delicate, they should be tested in a non-damaging manner. When it comes to fruit size, the most important physical feature is its hue, which gives the visual property. Automatic quality identification is performed utilizing a few images processing approaches that may be performed with the aid of picture characteristics such as shape, color, and size. This study focuses on image processing techniques such as segmentation and classification to differentiate between excellent and poor fruit.[10]
9	Plant disease identification using deep learning classification model: CNN	The author's goal in this study is to develop innovative technology for detecting plant diseases. They will first gather all photos using a camera or from an existing data set. They will next use a low pass filter to eliminate excess noise from the image and

		sharpen it. Following that, they will segment the picture using canny edge detection to determine the real edges of the leaf under detection. They then calculate the feature of the leaf, and if the feature of the test picture is comparable to any leaf in the training database, they predict that the leaf under detection has the same illness using a deep learning convolution neural network.[11]
10	Machine Learning-Based Detection and Grading of Varieties of Apples and Mangoes.	In the fruit grading process, a mostly manual technique has been employed, which has resulted in challenges with maintaining regularity, being time-consuming, and leaving the human operators exhausted. In the proposed project, they will create and implement a non-destructive mango grading system that is focused on deep learning. Collected mango fruits are used in the production of samples for the acquisition of RGB and thermal pictures, according to the technique. Using a computerized weighted scale, they able to determine the weight of the fruits. In the project, they design an

		autonomous picture capture system that will use both conventional and infrared cameras. Fruit grading quality is determined by the presence or absence of bruise, size, and maturity parameters.[12]
11	Automatic Date Fruit Recognition Using Outlier Detection Techniques and Gaussian Mixture Models	It has been suggested in this study that an automated date fruit identification technique been used. According on the percentage of fractures on the surface of the sample, Back Propagation Neural Networks (BPNN) are used to categorize date samples into three categories. During the processing of the date fruit, outlier samples are removed from the training samples of each variety in order to avoid a serious effect on the recognition results. The visual qualities of samples belonging to the same variety might change greatly amongst them even though they are of the same variety. Their suggested approach is capable of distinguishing between samples at various stages of development as well as between the very difficult to differentiate kinds. Aside from

		that, research findings indicated that the approach outperformed numerous other methods considered to be state of the art in terms of accuracy.[13]
12	A Transfer Learning-based Approach to Predict the Shelf life of Fruit	The motive of this paper is to identify the shelf-life of mangoes through a state-of-the-art non-destructive method, namely thermal imaging with three pre-trained models through transfer learning techniques. It has been determined that limited thumbnail datasets for training are required in order to achieve high accuracy in transfer learning. Food industry applications for shelf-life prediction of fruits based on visual inspection and RGB imaging via exterior characteristics are becoming increasingly common in agriculture and the food industry. However, by combining a thermal imaging technology with a deep learning-based classification strategy, it is possible to forecast shelf life, which has the potential to reduce food waste while also serving as a substitute for physical measures and human intervention.[14]

13	Fruit, vegetable and nut quality evaluation and control using computer vision	The purpose of this study is to present a complete assessment of current improvements in computer vision for inspecting and controlling the quality of nuts, fruits, and vegetables. Several image acquisition systems are explained, with a special emphasis on scene organization, lighting, and acquisition technology. Numerous industry criteria are crucial, including product singulation, real-time processing, and the ability to check the entire surface of the items. The article describes a case study of an actual inspection machine, which involved the creation of all the subsystems required to perform an automatic in-line inspection of a tiny, processed fruit. The growing multidisciplinary nature of research groups enables the integration of genetic, biological, and physiological information with physics and computer vision, paving the way for integrated solutions for the fruit and vegetable business. Internal product quality inspection will most likely be integrated into packing lines in the future utilizing one of

		the available technologies. This development is mostly due to the continuous decrease in the cost of machinery components and the increase in the processing power of current computers.[15]
14	Computer vision based automated billing system for fruit stores	The main aim of this work is to automate the billing procedure in fruit markets. An analogue weight sensor, an analogue to digital converter, a camera, a microcontroller unit, and a printer are included in the suggested system. The load cell is used to determine the weight of fruits that are stored in a basket. When the basket is placed on a weighing machine, the camera records an image of the fruits inside. To identify the sort of fruit, the collected image is preprocessed and evaluated using a deep learning algorithm. The computational technique in this study gets the measured weight of the fruit from the load cell, classifies the type of fruit using an image classifier program, and computes the total amount of fruits in the basket autonomously. The purpose of this

		article is to present the design and execution of a prototype real-time and resilient technique for automating fruit store billing systems. The work focuses on the recognition of fruits through the application of deep learning techniques and on the hardware design of weighing equipment. Once the recognition system's accuracy is increased, it may pave the way for a more dependable automated payment system for use in fruit markets.[16]
15	Image Segmentation K-Means Clustering Algorithm for Fruit Disease Detection Image Processing	Using an updated K-means method, this article attempts to divide the dataset into K pre-defined non-covering subgroups (bunches). It asks a collection of data that contains the collective of its models and is located closer to its center than to the convergence point of another collection. The system employs an easy approach to sorting a collection of a given instructive assortment into a specified number of groups. The E-K-Means clustering algorithm outperforms the other two. The Support Vector Machine

		algorithm has a confidence interval of 40.6 to 66.9, the Existing K-Means neighbor approach has a confidence interval of 49.6 to 77.5, and the E-K-Means Clustering algorithm has a confidence interval of 55 to 86.5. Each time, the E-K-Means clustering algorithm produces astounding results.[17]
16	Prospects of deep learning for medical imaging	This article demonstrates how machine learning techniques can be used to the medical field. Deep learning has emerged as a game-changing technology that has the potential to significantly improve the performance of conventional machine learning approaches and to tackle previously intractable issues. Medical imaging has been regarded as a critical area of research in which deep learning can make a big contribution. The purpose of this review article is to conduct a literature search on deep learning in medical imaging and to discuss its possibilities for future medical imaging research. In this research, the term CNN refers to a network architecture

		comprised of numerous layered convolutional layers. Medical image processing will greatly benefit from deep learning approaches, as deep learning has demonstrated exceptional performance in non-medical routine imaging studies. In this review paper, they touched on a brief history from conventional machine learning to deep learning, surveyed many deep learning applications in medical imaging, and concluded with limitations and future directions of deep learning in medical imaging. Despite these limitations, the benefits of DL greatly outweigh the drawbacks, and it will thus be a critical diagnostic tool tool in the era of precision medicine. [1] Drawback : If the comparison groups are significantly imbalanced, the deep learning algorithm will be unable to properly learn the under-represented group. Researchers should ensure that they have suitable subtype labels and an enough sample size for each subtype.

CHAPTER 3

METHODOLOGY

3.1 Project Design

Classification of the freshness of fruits based on the types of defects. Types of fruits that being chosen for this project is eggplant. When the deep learning algorithm is provided with 3 grades of eggplants. It is self-adjusted to classify (grade) the eggplant based on the sample sets. Every time an eggplant looks like one of the 3 grades, it is classified and graded accordingly. The algorithm is targeted to be able to evaluate fruit quality depending on parameters such as size, color, shape, and intensity, but in the same time color and size are the most important factors for fruit grading and sorting. Thus, this project is expected to build a system that can identify eggplant freshness automatically, which will save time, effort and provide better accuracy than Manual Sorting.

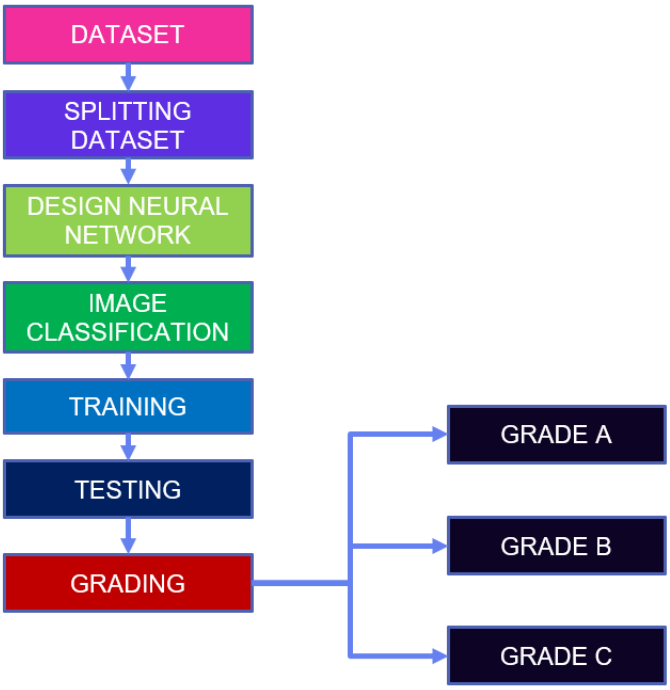


Figure 5: Project methodology flow chart

3.1.1 Collecting Dataset

Around 184 dataset of eggplants is collected from IOT based fertigation system for agriculture site at Satellite Station place in FKEKK, UTeM. It is a fertigation technology for agriculture that is based on the Internet of Things. According to the time, sensor data, and if requirements are met, the system will run automatically. The user will be able to monitor and manage the system both locally and remotely via the Internet of Things. This project is under Research and Innovation Management Center (CRIM), lead by Ir. Dr. Anas Bin Abdul Latiff and Dr Fakrulradzi Bin Idris.

Figure 6 : Eggplants

Figure 7 : Eggplants

Figure 8 : FKEKK Satellite

Figure 9 : CRIM IOT fertigation system

3.1.2 Design neural network

We will construct a deep neural network capable of picture recognition. Deep learning excels in recognizing objects in pictures because it is implemented using three or more layers of artificial neural networks, each of which is responsible for extracting one or more image features (more on that later). A neural network is a computational model that operates in the same manner as neurons in the human brain do. Each neuron receives an input, conducts certain processes, and then sends the output to the next neuron. The application will be written in Python with TensorFlow. TensorFlow is a Google open-source deep learning framework that provides developers with granular control over each neuron (known as a "node" in TensorFlow) so that weights may be adjusted to achieve optimal performance.

3.1.3 Splitting Dataset

The dataset contains 3 classes. Firstly, it needs to properly prepare the dataset, as this is the initial step in solving any machine learning challenge. To split the dataset, we must store our data in the predetermined directory structure shown below; we only need to place the photos in the appropriate class folder.

Figure 10: Example image of eggplants from dataset

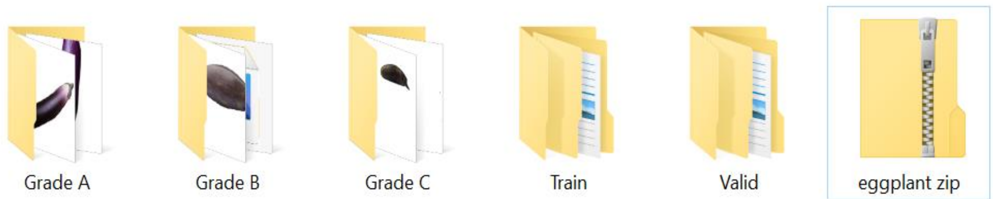


Figure 11: Dataset

3.1.4 Image Classification

Image classification is a complex procedure which relies on different components. Image categorization is crucial in remote sensing photos and is utilized for a variety of applications including environmental change, agriculture, land use/land planning, urban planning, surveillance, geographic mapping, disaster control, and object identification.

3.1.4.1 Image Acquisition and Pre-processing

First, a picture of the fruit is captured using any image capturing equipment. To get a new brightness value in the output picture, preprocessing methods employ a tiny neighborhood of a pixel in an input image. Filtration is another term for such preprocessing processes. Local preprocessing methods are classified into two types based on their processing goal: smoothing suppressed noise or other tiny variations in the image; equal to suppressing high frequencies in the frequency range.[9]

I. Importing a picture with the use of image receiving tools.

II. Performing picture analysis and manipulation, such as data compression, image enhancement, and so on.

III. Outputs that may be adjusted depending on image analysis include images, reports, and other documents.

3.1.4.2 Feature Extraction

The process of mapping image pixels into the feature space is known as feature extraction. For objects to be automatically recognized from remote sensing data, they must have particular attributes that define and distinguish them from one another.

Feature extraction aids in the reduction of redundant data in a data collection. Finally, data reduction helps the machine to build the model with less machine efforts while also speeding up the learning and generalization phases of the machine learning process.

3.1.5 Training Dataset

3.1.5.1 Image Augmentation

The method of image augmentation is the process of forming images for training the deep learning model. These new images are created by reusing previously saved training images. Augmentation is the action or process of increasing in size or quantity. Deep learning model require a big amount of training data to generalize well and attain high accuracy in deep learning. However, in some circumstances, image data is insufficiently large enough. In this situation, a strategy is used to augment the training data. This technique generates training data artificially by processing the input data with techniques such as random rotation, shifts, shear, and flip.

3.1.5.2 Image Augmentation With ImageDataGenerator

At each epoch, the ImageDataGenerator class guarantees that the model receives new variations of the images. However, it merely returns the modified images and does not merge them into the original image corpus. If this is the case, the model will be exposed to the original photos many times, which will cause the model to overfit. However, ImageDataGenerator consumes less memory. This is because if this class is not used, it will load all the photos simultaneously. However, when it is used, the photos are loaded in batches, which saves a lot of memory.

3.1.6 Testing and Grading

In order to test the network, each of the pictures in the testing set must be shown to the network and asked to guess what the label of the image it believes to be. It is then necessary to tabulate the model's predictions for each image in the testing set.

Finally, these model predictions are compared to the ground-truth labels from the testing set. When you look at the ground truth labels, you can see exactly what the picture category is. From there, it is possible to calculate the number of accurate predictions made by the classifier, as well as aggregate reports such as precision, recall, and f-measure, which are used to characterize the overall performance of the network as a whole.

3.2 Project Testing

3.2.1 Image classification using 2 class

Based on the classification of cat and dog projects as examples Attempts have been made by using the eggplant dataset in this study. However, it can only be classified into two types of eggplant: grade A and grade B.

3.2.1.1 Building convolutional neural network

First, a neural network must be constructed. The Sequential model from Keras, together with the required packages for convolutional neural network stacking, is utilized since the network is a sequential of layers.

```
import tensorflow as tf
from keras.preprocessing.image import ImageDataGenerator
from PIL import Image
```

Figure 12 : Import TensorFlow

The input shape parameters have been configured, and the activation function has been set to rectifier. For the 2D array, 32 features and a 3x3 array is used. All of the images will be converted to a 64x64 3D array (Since it is in color). And pooling is used to reduce the size of the feature map obtained during the convolution step. (Divide by 2). For a more deep cognitive network, a second layer is required.


```
cnn = tf.keras.models.Sequential()
cnn.add(tf.keras.layers.Conv2D(filters=32, kernel_size=3, activation='relu', input_shape=[64, 64, 3]))
cnn.add(tf.keras.layers.MaxPool2D(pool_size=2, strides=2))
cnn.add(tf.keras.layers.Conv2D(filters=32, kernel_size=3, activation='relu'))
cnn.add(tf.keras.layers.MaxPool2D(pool_size=2, strides=2))
cnn.add(tf.keras.layers.Flatten())
cnn.add(tf.keras.layers.Dense(units=128, activation='relu'))
cnn.add(tf.keras.layers.Dense(units=1, activation='sigmoid'))
```

```
cnn.compile(optimizer = 'adam', loss = 'binary_crossentropy', metrics = ['accuracy'])
```

Figure 13: Sequential Model

The 2D pooled array is flatten into a vector. In order for anything to work, everything must be connected. The probability of a picture having terung A or B may be computed using a rectifer function, followed by a sigmoid function.

After that, the neural network is put together. This neural network is built with help of the Adam optimizer, an effective approach for first-order gradient-based optimization of stochastic objective functions that is based on adaptive estimations of lower-order moments. The Adam optimizer is used to construct the neural network. It is necessary to employ a log loss function and binary cross entropy. Because it has a good relationship with the sigmoid function. The accuracy of the model will be measured, since this is what the model's developers are most concerned about.

3.2.1.2 Fitting the Convolutional Neural Network to the Images

For each epoch, a total of 86 images are selected for the training set. In addition, 57 images have been selected for the validation procedures.


```
train_datagen = ImageDataGenerator(rescale = 1./255,
                                   shear_range = 0.2,
                                   zoom_range = 0.2,
                                   horizontal_flip = True)
training_set = train_datagen.flow_from_directory('/home/adila/Documents/contoh/terung train',
                                                 target_size = (64, 64),
                                                 batch_size = 32,
                                                 class_mode = 'binary')
```

```
Found 86 images belonging to 3 classes.
```

```
test_datagen = ImageDataGenerator(rescale = 1./255)
test_set = test_datagen.flow_from_directory('/home/adila/Documents/contoh/terung valid',
                                            target_size = (64, 64),
                                            batch_size = 32,
                                            class_mode = 'binary')
```

```
Found 57 images belonging to 3 classes.
```

Figure 14 : Training set and Validation set

```
cnn.fit(x = training_set, validation_data = test_set, epochs = 25)

Epoch 1/25
3/3 [==============================] - 69s 26s/step - loss: -0.1554 - accuracy: 0.3372 - val_loss: -2.1130 - val_acc
uracy: 0.2807
Epoch 2/25
3/3 [==============================] - 42s 18s/step - loss: -2.6432 - accuracy: 0.3372 - val_loss: -5.4782 - val_acc
uracy: 0.2807
Epoch 3/25
3/3 [==============================] - 50s 20s/step - loss: -5.3982 - accuracy: 0.3372 - val_loss: -11.4994 - val_ac
curacy: 0.2807
Epoch 4/25
3/3 [==============================] - 44s 18s/step - loss: -12.4016 - accuracy: 0.3372 - val_loss: -21.9139 - val_a
ccuracy: 0.2807
Epoch 5/25
3/3 [==============================] - 42s 17s/step - loss: -20.3196 - accuracy: 0.3372 - val_loss: -40.5791 - val_a
ccuracy: 0.2807
Epoch 6/25
3/3 [==============================] - 47s 19s/step - loss: -40.9124 - accuracy: 0.3372 - val_loss: -64.0022 - val_a
ccuracy: 0.2807
Epoch 7/25
3/3 [==============================] - 63s 28s/step - loss: -59.6237 - accuracy: 0.3372 - val_loss: -100.1473 - val_
accuracy: 0.2807
```

Figure 15 : Epoch

3.2.1.3 Making prediction

Finally, the neural network is tested by entering a random photo from Google to check whether it can accurately recognize and classify the image that has been inserted.

```
In [13]: import numpy as np
         from keras.preprocessing import image
         test_image = image.load_img('/home/adila/Documents/contoh/test/index3.jpeg', target_size = (64, 64))
         test_image = image.img_to_array(test_image)
         test_image = np.expand_dims(test_image, axis = 0)
         result = cnn.predict(test_image)
         training_set.class_indices
         if result [0][0] == 1:
           prediction = 'terung B'
         else :
           prediction = 'terung A'

         display(image.load_img('/home/adila/Documents/contoh/test/index3.jpeg'))
```

```
In [14]: prediction
Out[14]: 'terung A'
```

Figure 16 : Prediction for eggplant grade A

```
In [17]: import numpy as np
         from keras.preprocessing import image
         test_image = image.load_img('/home/adila/Documents/contoh/test/contoh.jpeg', target_size = (64, 64))
         test_image = image.img_to_array(test_image)
         test_image = np.expand_dims(test_image, axis = 0)
         result = cnn.predict(test_image)
         training_set.class_indices
         if result [0][0] == 1:
           prediction = 'terung B'
         else :
           prediction = 'terung A'

         display(image.load_img('/home/adila/Documents/contoh/test/contoh.jpeg'))
```

```
In [18]: prediction
Out[18]: 'terung B'
```

Figure 17 : Prediction for eggplant grade B

3.2.2 Image classification using google collab

3.2.2.1 Load the data

Eggplant dataset is load from google drive into google collab.


```
[1] import tensorflow as tf
    from tensorflow import keras
    from tensorflow.keras import layers
```

```
[3] from zipfile import ZipFile
    file_name = "/content/drive/MyDrive/terung/terung valid.zip"

    with ZipFile(file_name, 'r') as zip:
      zip.extractall()
      print("Done")

    Done
```

Figure 18 : Load the data

3.2.2.2 Generate a Dataset

The image from the train and validation directories is now being processed into the dataset.

```
[5] image_size = (180, 180)
    batch_size = 32

    train_ds = tf.keras.preprocessing.image_dataset_from_directory(
        "/content/terung train",
        validation_split=0.2,
        subset="training",
        seed=1337,
        image_size=image_size,
        batch_size=batch_size,
    )
    val_ds = tf.keras.preprocessing.image_dataset_from_directory(
        "/content/terung valid",
        validation_split=0.2,
        subset="validation",
        seed=1337,
        image_size=image_size,
        batch_size=batch_size,
    )

    Found 86 files belonging to 3 classes.
    Using 69 files for training.
    Found 57 files belonging to 3 classes.
    Using 11 files for validation.
```


Figure 19 : Generate a Dataset

3.2.2.3 Visualize the data

The first six images from the training dataset will be shown using this code. Label 0 indicates "grade A," label 1 indicates "grade B," and label 2 indicates "grade C."

```
import matplotlib.pyplot as plt

plt.figure(figsize=(10, 10))
for images, labels in train_ds.take(1):
    for i in range(6):
        ax = plt.subplot(3, 3, i + 1)
        plt.imshow(images[i].numpy().astype("uint8"))
        plt.title(int(labels[i]))
        plt.axis("off")
```

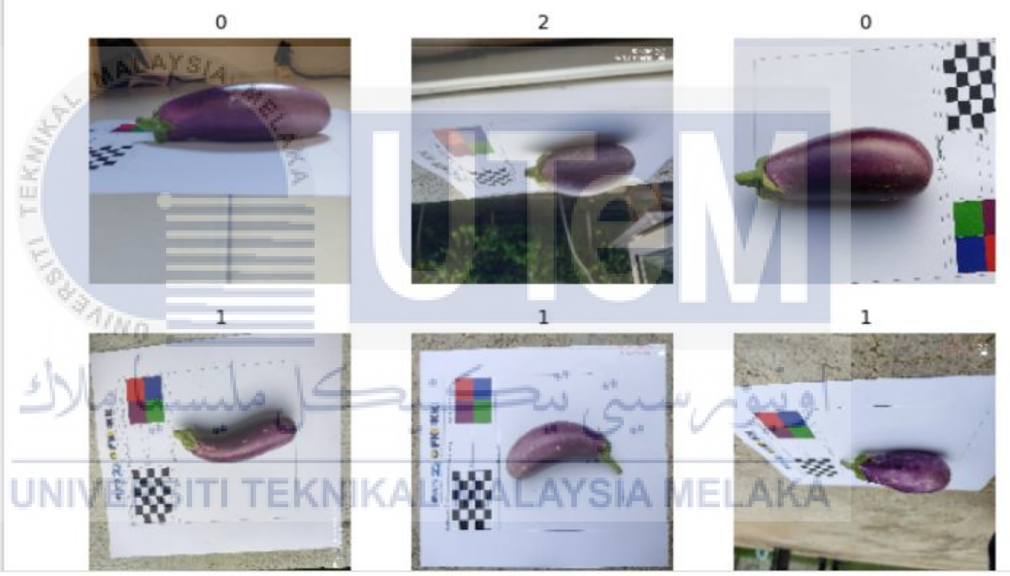


Figure 20 : Visualize the data

3.2.2.4 Using image data augmentation

The approach of introducing random but realistic modifications to the training pictures, such as random horizontal flips or minor random rotations, is recommended when the image collection is limited. This helps to expose the model to diverse elements of the training data while also slowing down overfitting.

```
[8] plt.figure(figsize=(10, 10))
    for images, _ in train_ds.take(1):
        for i in range(9):
            augmented_images = data_augmentation(images)
            ax = plt.subplot(3, 3, i + 1)
            plt.imshow(augmented_images[0].numpy().astype("uint8"))
            plt.axis("off")
```

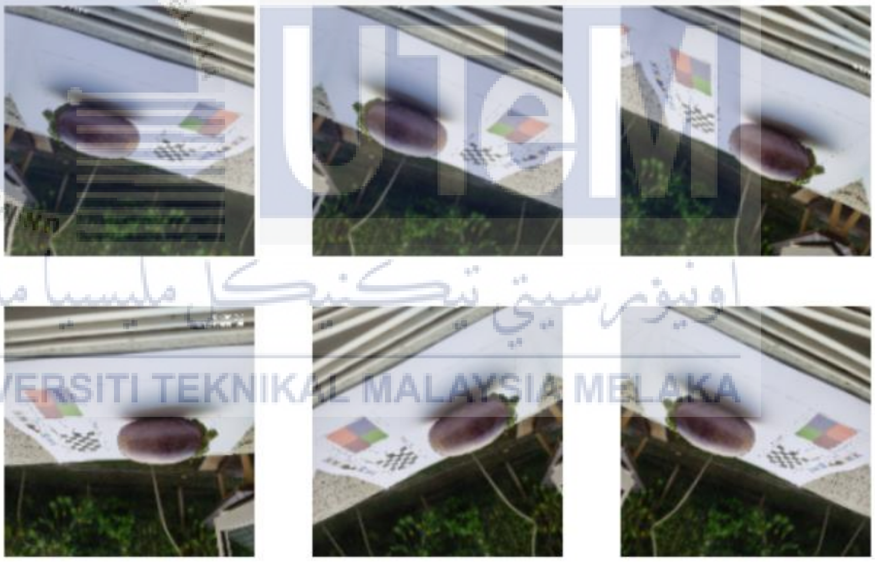


Figure 21 : Image Data Augmentation

Preferably, use buffered prefetching so that data may be retrieved from disc without causing I/O to get blocked.

```
[11] train_ds = train_ds.prefetch(buffer_size=32)
     val_ds = val_ds.prefetch(buffer_size=32)
```

Figure 22 : Configure the dataset for performance

3.2.2.5 Build a model

A scaled-down version of the Xception network is being created. It starts with the model and the data augmentation preprocessor, and then moves on to the Rescaling layer and finally the final layer. As an extra precaution, insert a Dropout layer before the final classification layer.

```
def make_model(input_shape, num_classes):
    inputs = keras.Input(shape=input_shape)
    # Image augmentation block
    x = data_augmentation(inputs)

    # Entry block
    x = layers.Rescaling(1.0 / 255)(x)
    x = layers.Conv2D(32, 3, strides=2, padding="same")(x)
    x = layers.BatchNormalization()(x)
    x = layers.Activation("relu")(x)

    x = layers.Conv2D(64, 3, padding="same")(x)
    x = layers.BatchNormalization()(x)
    x = layers.Activation("relu")(x)

    previous_block_activation = x  # Set aside residual

    for size in [128, 256, 512, 728]:
        x = layers.Activation("relu")(x)
        x = layers.SeparableConv2D(size, 3, padding="same")(x)
        x = layers.BatchNormalization()(x)

        x = layers.Activation("relu")(x)
        x = layers.SeparableConv2D(size, 3, padding="same")(x)
        x = layers.BatchNormalization()(x)

        x = layers.MaxPooling2D(3, strides=2, padding="same")(x)

        # Project residual
        residual = layers.Conv2D(size, 1, strides=2, padding="same")(
            previous_block_activation
        )
        x = layers.add([x, residual])  # Add back residual
        previous_block_activation = x  # Set aside next residual

    x = layers.SeparableConv2D(1024, 3, padding="same")(x)
    x = layers.BatchNormalization()(x)
    x = layers.Activation("relu")(x)

    x = layers.GlobalAveragePooling2D()(x)
    if num_classes == 3:
        activation = "sigmoid"
        units = 1
    else:
        activation = "softmax"
        units = num_classes

    x = layers.Dropout(0.5)(x)
    outputs = layers.Dense(units, activation=activation)(x)
    return keras.Model(inputs, outputs)


model = make_model(input_shape=image_size + (3,), num_classes=3)
keras.utils.plot_model(model, show_shapes=True)
```


Figure 23 : Build a model

3.2.2.6 Train the model

The model that has been constructed is being trained over a period of 50 epochs.

```
epochs = 50

callbacks = [
    keras.callbacks.ModelCheckpoint("save_at_{epoch}.h5"),
]
model.compile(
    optimizer=keras.optimizers.Adam(1e-3),
    loss="binary_crossentropy",
    metrics=["accuracy"],
)
model.fit(
    train_ds, epochs=epochs, callbacks=callbacks, validation_data=val_ds,
)
```

Figure 24 : Train the model

3.2.2.7 Run inference on new data

Finally, the neural network is evaluated by inserting a random picture from Google to see whether it can properly detect and categorize the image that has been added. The percent of similarity between the image and the image in the dataset is shown to determine accuracy.


```
[20] img = keras.preprocessing.image.load_img(
         "/content/drive/MyDrive/contoh terung/images.jfif", target_size=image_size
     )
     img_array = keras.preprocessing.image.img_to_array(img)
     img_array = tf.expand_dims(img_array, 0)  # Create batch axis

     predictions = model.predict(img_array)
     score = predictions[0]
     print(
         "This image is %.2f percent terung A, %.2f percent terung B and %.2f percent terung C."
         % (100 * (1 - score), 100 * (1 - score), 100 * score)
     )

This image is 0.00 percent terung A, 0.00 percent terung B and 100.00 percent terung C.
```

Figure 25 : Grading

CHAPTER 4

RESULTS AND DISCUSSION

The suggested system is built in Jupyter (python 3.7), and the eggplant dataset that was utilized comprises of photos in the grade A, grade B, and grade C categories in the.png format. Results are acquired by applying the suggested feature extraction methods to the eggplant dataset and observing the outcomes. The main concern of this study is to automatically recognize different grade of eggplant using deep learning techniques. In this section, details are provided about the project result.

4.1 Designed Model

Image for dataset from 148 images was collected. 122 images will be used for model training (and will be saved in the train folder), and 26 images will be used for model validation (and will be stored in the validate folder). There are three files in each matching folder, each having image of eggplant in the grade A, grade B, and grade C categories.

The ImageDataGenerator class will be used in the data augmentation process to generate images. CNN architecture will be defined by the functions Conv2D, MaxPool2D, Flatten, Dense, and InputLayer. In the data augmentation process, the ImageDataGenerator class is utilized. Through a series of adjustments to original pictures, such as rotation, shifting, flipping, and shearing, this approach will significantly increase the amount of training data available. The generalizability of the model will also be improved as a result of this strategy.

```
train_datagen = ImageDataGenerator(rescale= 1./255, validation_split= 0.5, shear_range = 0.2, zoom_range= 0.2, rotati
validation_datagen = ImageDataGenerator(rescale=1./255, validation_split= 0.5)
```

```
train_generator = train_datagen.flow_from_directory(
            training_dir,
            target_size=(100,100),
            batch_size=5,
            class_mode='categorical',
            shuffle=True
)
valid_generator = validation_datagen.flow_from_directory(
            validation_dir,
            target_size=(100,100),
            batch_size=5,
            class_mode='categorical',
            shuffle=True,
            subset='validation'
)
```

```
Found 122 images belonging to 3 classes.
Found 26 images belonging to 3 classes.
```

Figure 26 : ImageDataGenerator

The architecture of CNN has been constructed. The model was then run using a variety of batch sizes and epochs to allow it to be compared to other models in the study. The sequential model developed by Keras is used to construct the model. Each layer of this empty model is added one at a time until the model is complete. In the first layer, there is a convolutional layer with a depth of 16 and an activation of 'relu'.

Due to the fact that following layers may deduce their input sizes from their predecessors' output sizes, the input size is specified only once, at the beginning of the first layer. The supplied data set is of size (100, 100, 3). A MaxPooling2D layer is added after each convolutional layer in the pipeline. The largest value over a window is used to sample the input representation in this layer down sampling technique. Pooling is essentially the act of integrating data with the goal of lowering the overall size of the data set. Finally, a Flatten and a Dropout layer is added to the model (with a probability of 0.25), and the model is complete. A 1-dimensional array is created by flattening the data, which is then passed on to the following layer. Dropout is a technique used in neural networks to minimize overfitting. The activation function for the last layer is 'softmax.' This layer is the most complex.

```
model = Sequential()

model.add(Conv2D(16, kernel_size=3, activation='relu', input_shape=(100, 100, 3)))
model.add(MaxPool2D(2,2))
model.add(Dropout(0.25))
model.add(Conv2D(32, kernel_size=3, activation='relu'))
model.add(MaxPool2D(2,2))
model.add(Dropout(0.25))
model.add(Conv2D(64, kernel_size=3, activation='relu'))
model.add(MaxPool2D(2,2))
model.add(Dropout(0.25))
model.add(Conv2D(128, kernel_size=3, activation='relu'))
model.add(MaxPool2D(2,2))
model.add(Dropout(0.25))
model.add(Flatten())
model.add(Dense(512, activation='relu'))
model.add(Dense(3, activation='softmax'))

model.summary()
```

Figure 27 : Sequential Model

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_4 (Conv2D)           (None, 98, 98, 16)        448

 max_pooling2d_4 (MaxPooling  (None, 49, 49, 16)       0
 2D)

 dropout_4 (Dropout)         (None, 49, 49, 16)        0

 conv2d_5 (Conv2D)           (None, 47, 47, 32)        4640

 max_pooling2d_5 (MaxPooling  (None, 23, 23, 32)       0
 2D)

 dropout_5 (Dropout)         (None, 23, 23, 32)        0

 conv2d_6 (Conv2D)           (None, 21, 21, 64)        18496

 max_pooling2d_6 (MaxPooling  (None, 10, 10, 64)       0
 2D)

 dropout_6 (Dropout)         (None, 10, 10, 64)        0

 conv2d_7 (Conv2D)           (None, 8, 8, 128)         73856

 max_pooling2d_7 (MaxPooling  (None, 4, 4, 128)        0
 2D)

 dropout_7 (Dropout)         (None, 4, 4, 128)         0

 flatten_1 (Flatten)         (None, 2048)              0

 dense_2 (Dense)             (None, 512)               1049088

 dense_3 (Dense)             (None, 3)                 1539

=================================================================
Total params: 1,148,067
Trainable params: 1,148,067
Non-trainable params: 0
_________________________________________________________________
```


Figure 28 : Sequential Model Summary

4.2 Analysis

A number of analyses were conducted in order to determine the accuracy of the model. It has been attempted to use various values of epoch and batch sizes.

4.2.1 Batch size

The results of different batch sizes were shown in the table for comparison purposes. Because of the minimal quantity of data in the dataset, a smaller batch size is used. The greater the number of batches, the greater the probability of poor generalizations. However, the bigger the number of batch sizes, the faster the model per epoch throughout the training process.

Table 3 : Batch size

Batch size	Percentage%	
	Train accuracy	Valid accuracy
3	84%	69%
4	86%	84%
5	79%	84%
6	79%	84%
7	77%	50%

4.2.2 Epoch

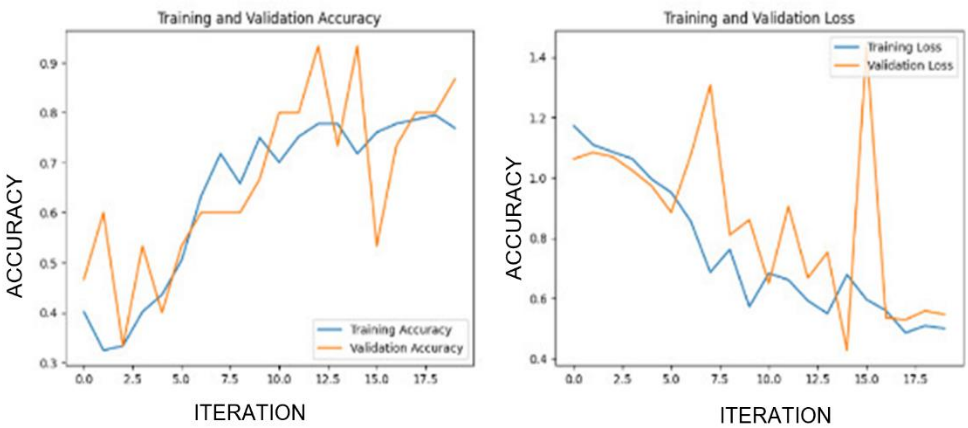
For comparison purposes, various amounts of epoch are also tested and displayed in the table. Because it has a greater percentage validation accuracy than the other epochs in the table, epoch 50 appears to be the most appropriate one. An overwhelming number of epochs may lead the model to over-fit the training data in general. Therefore, the model does not learn the data; rather, it memorizes the data

Table 4 : Epoch

Epoch	Percentage%	
	Train accuracy	Valid accuracy
10	75%	80%
20	75%	80%
30	75%	80%
40	72%	80%
50	85%	84%

4.2.3 Graph

The batch size of 4 and the epoch of 50 were determined to be the most appropriate values. The accuracy of the model's trains is 85.25 percent on average. Validation accuracy, on the other hand, is 84.62 percent. Accuracy and loss in each epoch are plotted in order to be able to examine the model learning process. As the epoch value increases, it is expected that loss values will drop and accuracy values will increase, respectively. The accuracy graphs for training and validation, as well as the loss graphs for training and validation, were presented in the picture.



4.3 Result

The model will attempt to identify the appearance of eggplants. For each class, this function returns an output array containing the probability for that class. If the picture has a value equal to 0, it is considered Grade A. If the picture has a value of 1, the Grade is B. If the picture has a value of 2, it is considered Grade C.

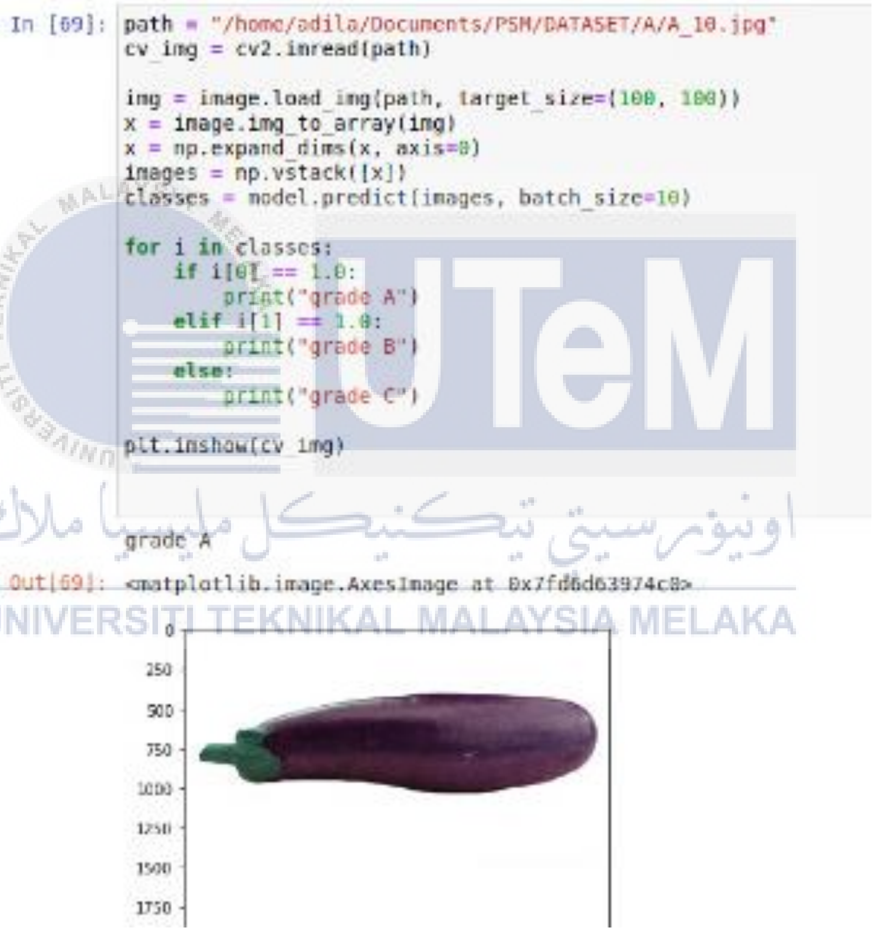


Figure 29 : Result for Grade A eggplant

```
In [68]: path = "/home/adila/Documents/PSM/DATASET/B/B_01.png"
         cv_img = cv2.imread(path)

         img = image.load_img(path, target_size=(100, 100))
         x = image.img_to_array(img)
         x = np.expand_dims(x, axis=0)
         images = np.vstack([x])
         classes = model.predict(images, batch_size=10)

         for i in classes:
             if i[0] == 1.0:
                 print("grade A")
             elif i[1] == 1.0:
                 print("grade B")
             else:
                 print("grade C")

         plt.imshow(cv_img)

         grade B

Out[68]: <matplotlib.image.AxesImage at 0x7fd6d642c580>
```

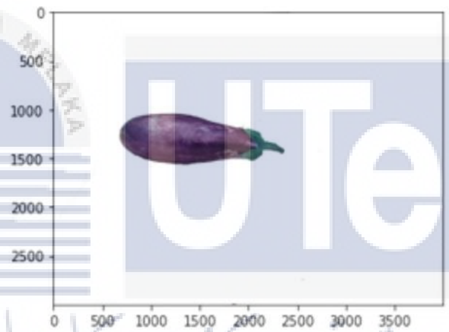


Figure 30 : Result for Grade B eggplant

```
In [65]: path = "/home/adila/Documents/PSM/DATASET/C/C_50.png"
         cv_img = cv2.imread(path)

         img = image.load_img(path, target_size=(100, 100))
         x = image.img_to_array(img)
         x = np.expand_dims(x, axis=0)
         images = np.vstack([x])
         classes = model.predict(images, batch_size=10)

         for i in classes:
             if i[0] == 1.0:
                 print("grade A")
             elif i[1] == 1.0:
                 print("grade B")
             else:
                 print("grade C")

         plt.imshow(cv_img)

         grade C

Out[65]: <matplotlib.image.AxesImage at 0x7fd6c8cd3b50>
```

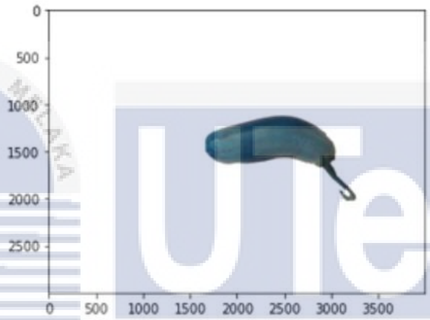


Figure 31 : Result for Grade C eggplants

CHAPTER 5

CONCLUSION AND FUTURE WORKS

As a result of the findings and discussions presented in Chapter 4, the conclusion and future research will be discussed in this chapter (Result and Discussion).

5.1 Conclusion

The classification of crops using deep learning technique is proposed in this study. In order to acquire the results, a total of 148 datasets were gathered from the IOT agricultural project under the CRIM for training and validation. The algorithm was coded and tested with the help of the jupyter notebook software. A test is conducted in Chapter 4 (Results and Discussion) to achieve the second objective that is to validate the performance of the classification technique of the Deep Learning model in terms of accuracy, through the test the several number of batch size and epoch is used to the constructed model, which is then used to determine the optimal value for batch size and epoch that is most convenient with the dataset. This is due to the fact that selecting the appropriate batch size and epoch to fit in extremely small datasets is critical in determining the overall performance of the model. By experimenting with different combinations of hidden layers, the accuracy and loss curves were created. In addition, this project discusses several approaches and algorithms for crop categorization that are based on computer vision techniques. Improved performance of CNN in order to get better fruit categorization. The goal of the designed model is to classify eggplant grades into three categories: Grade A, Grade B, and Grade C. The third objective is achieved when the proposed technique achieved an accuracy rate of 85 %. In the conclusion all three objective for this project has been achieved.

5.2 Future work

However, in future projects, I hope to increase this research using more dataset that includes several grading systems for various types of fruit and vegetables. I propose to develop many other CNN-based models to compare their accuracy on the same dataset. Furthermore, additional characteristics for grading and categorization can be developed to help identify disease kinds and/or fruit texture structure.

REFERENCES

[1] J. Kim, J. Hong, and H. Park, “Prospects of deep learning for medical imaging,” *Precis. Futur. Med.*, vol. 2, no. 2, pp. 37–52, 2018, doi: 10.23838/pfm.2018.00030.

[2] W. I. W. Ishak and R. M. Hudzari, “Image based modeling for oil palm fruit maturity prediction,” *J. Food, Agric. Environ.*, vol. 8, no. 2, pp. 469–476, 2010.

[3] Harsawardana *et al.*, “AI-Based Ripeness Grading for Oil Palm Fresh Fruit Bunch in Smart Crane Grabber,” in *IOP Conference Series: Earth and Environmental Science*, Mar. 2020, vol. 426, no. 1, doi: 10.1088/1755-1315/426/1/012147.

[4] S. Aich and I. Stavness, “Leaf counting with deep convolutional and deconvolutional networks,” in *Proceedings - 2017 IEEE International Conference on Computer Vision Workshops, ICCVW 2017*, 2017, vol. 2018-Janua, doi: 10.1109/ICCVW.2017.244.

[5] A. K. Rangarajan, R. Purushothaman, and A. Ramesh, “Tomato crop disease

classification using pre-trained deep learning algorithm," *Procedia Comput. Sci.*, vol. 133, pp. 1040–1047, 2018, doi: 10.1016/j.procs.2018.07.070.

[6] M. B. Villanueva and M. L. M. Salenga, "Bitter melon crop yield prediction using Machine Learning Algorithm," *Int. J. Adv. Comput. Sci. Appl.*, vol. 9, no. 3, pp. 1–6, 2018, doi: 10.14569/IJACSA.2018.090301.

[7] N. Agila and P. Senthil Kumar, "An efficient crop identification using deep learning," *Int. J. Sci. Technol. Res.*, vol. 9, no. 1, pp. 2805–2808, 2020.

[8] V. K. Shrivastava, M. K. Pradhan, S. Minz, and M. P. Thakur, "Rice plant disease classification using transfer learning of deep convolution neural network," *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci. - ISPRS Arch.*, vol. 42, no. 3/W6, pp. 631–635, 2019, doi: 10.5194/isprs-archives-XLII-3-W6-631-2019.

[9] S. Naik and B. Patel, "Machine Vision based Fruit Classification and Grading - A Review," *Int. J. Comput. Appl.*, vol. 170, no. 9, pp. 22–34, 2017, doi: 10.5120/ijca2017914937.

[10] A. Harsh, K. K. Jha, S. Srivastava, A. Raj, and S. Raghav, "Fruit Freshness Detection Using Cnn Approach," no. 06, pp. 456–463, 2020.

[11] S. Mangal and P. Meshram, "Plant Disease Identification Using Deep Learning Classification Model : Cnn," *J. Univ. Shanghai Sci. Technol.*, vol. 23, no. 1, pp. 253–263, 2021, [Online]. Available: https://jusst.org/wp-content/uploads/2021/01/PLANT-DISEASE-IDENTIFICATION-USING-DEEP-LEARNING-CLASSIFICATION-MODEL-CNN-1.pdf.

[12] V. Bhole and A. Kumar, “Mango Quality Grading using Deep Learning Technique: Perspectives from Agriculture and Food Industry,” *SIGITE 2020 - Proc. 21st Annu. Conf. Inf. Technol. Educ.*, pp. 180–186, 2020, doi: 10.1145/3368308.3415370.

[13] O. Aiadi, M. L. Kherfi, and B. Khaldi, “Automatic date fruit recognition using outlier detection techniques and Gaussian mixture models,” *Electron. Lett. Comput. Vis. Image Anal.*, vol. 18, no. 1, pp. 52–75, 2019, doi: 10.5565/rev/elcvia.1041.

[14] V. Bhole and A. Kumar, “A transfer learning-based approach to predict the shelf life of fruit,” *Intel. Artif.*, vol. 24, no. 67, pp. 102–120, 2021, doi: 10.4114/intartif.vol24iss67pp102-120.

[15] J. Blasco, N. Aleixos, S. Cubero, and D. Lorente, *Fruit, vegetable and nut quality evaluation and control using computer vision*. Woodhead Publishing Limited, 2012.

[16] M. Sugadev, K. Sucharitha, I. R. Sheeba, and B. Velan, “Computer vision based automated billing system for fruit stores,” *Proc. 3rd Int. Conf. Smart Syst. Inven. Technol. ICSSIT 2020*, no. Icssit, pp. 1337–1342, 2020, doi: 10.1109/ICSSIT48917.2020.9214101.

[17] P. K. Devi and Rathamani, “Image Segmentation K-Means Clustering Algorithm for Fruit Disease Detection Image Processing,” *Proc. 4th Int. Conf. Electron. Commun. Aerosp. Technol. ICECA 2020*, pp. 861–865, 2020, doi: 10.1109/ICECA49313.2020.9297462.