CLASSIFICATION AND ANALYSIS THE MATURITY AND CLASS VARIETY OF CHILI FRUIT USING CONVOLUTIONAL NEURAL NETWORK (CNN)

NAJIHAH BINTI MOHD HUSSIN

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

CLASSIFICATION AND ANALYSIS THE MATURITY AND CLASS VARIETY OF CHILI FRUIT USING CONVOLUTIONAL NEURAL NETWORK (CNN)

NAJIHAH BINTI MOHD HUSSIN

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

Faculty of Electronic and Computer Engineering
Universiti Teknikal Malaysia Melaka

2022

DECLARATION

I declare that this report entitled “Classification and Analysis the Maturity and Class Variety of Chili Fruit using Convolutional Neural Network (CNN)” results from my work except for quotes cited in the references.

Signature : …………………………………

Author : NAJIHAH BINTI MOHD HUSSIN

Date : 10 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 : TS. DR MUHAMMAD NOORAZLAN SHAH BIN ZAINUDIN

Date : 10 JANUARY 2022

DEDICATION

I am particularly grateful for the support and encouragement of my loving family and friends. Thanks to my supervisor, Ts. Dr Muhammad Noorazlan Shah Bin Zainudin and all of my lecturers, for their help and advice in completing my final year project, " Classification and Analysis the Maturity and Class Variety of Chili Fruit using Convolution Neural Network (CNN) ".

ABSTRACT

Autonomous robots have recently grown in popularity in the agriculture industry, although they are still unable to do tasks as well as humans. A thorough procedure produces erroneous results, which raises the cost of production. This study presents an alternative way for farmers who need to sort the fruit categories such as chilis according to tehir maturities. This kind of innovation could be tackled by adopting an Artificial Intelligence (AI) approach such as deep learning. Farmers can save labour expenses while increasing fruit harvests by implementing an innovative chili identification method. A total of 1200 256 X 256 pixel chilli images are used, with 840 being used for training and the remaining 360 being served for testing. In this experiment, a learning rate of 0.0001 is used, with a mini batch size of 64 and 400 iterations. Convolutional Neural Network (CNN) model is applied to learn and recognize the chili fruits into three categories; unmatured, moderately mature, and mature. Chili fruit maturity is determined by measuring the distance between the calyx and the apex. In addition,the variety class of chili fruits is also being recognized according to its colors into green, red, and overripe chili. Using ADAM and SGDM optimizers with multiple CNN architectures, this study is capable of recognising and classifying chilli fruits with an accuracy of above 85%. We intend to expand the

experiment with 3D images in the future, considering the depth information of the images in developing an autonomous agriculture robot.

ABSTRAK

Robot autonomi baru-baru ini semakin popular dalam industri pertanian, walaupun mereka masih tidak dapat melakukan tugas seperti manusia. Prosedur yang teliti menghasilkan keputusan yang salah, yang meningkatkan kos pengeluaran. Kajian ini mengemukakan cara alternatif kepada petani yang perlu menyusun kategori buah seperti cili mengikut kematangan tehir. Inovasi seperti ini boleh ditangani dengan menggunakan pendekatan Kecerdasan Buatan (AI) seperti pembelajaran mendalam. Petani boleh menjimatkan perbelanjaan buruh sambil meningkatkan hasil tuaian buah-buahan dengan melaksanakan kaedah pengenalan cili yang inovatif. Sebanyak 1200 256 X 256 piksel imej cili digunakan, dengan 840 digunakan untuk latihan dan baki 360 dihidangkan untuk ujian. Dalam eksperimen ini, kadar pembelajaran 0.0001 digunakan, dengan saiz kelompok mini 64 dan 400 lelaran. Model Convolutional Neural Network (CNN) digunakan untuk mempelajari dan mengenali buah cili kepada tiga kategori; belum matang, sederhana matang, dan matang. Kematangan buah cili ditentukan dengan mengukur jarak antara kelopak dan puncak. Selain itu, kelas varieti buah cili turut dikenali mengikut warnanya kepada cili hijau, merah dan terlalu masak. Menggunakan pengoptimum ADAM dan SGDM dengan berbilang seni bina CNN, kajian ini mampu mengecam dan *mengklasifikasikan buah cili dengan ketepatan melebihi 85%. Kami berhasrat untuk mengembangkan percubaan dengan imej 3D pada masa hadapan, dengan mengambil kira maklumat kedalaman imej dalam membangunkan robot pertanian autonomi.*

ACKNOWLEDGEMENTS

First of all, I would like to express deep gratitude to Allah SWT to achieve my project. I successfully finish this research on time, even though I had a lot of problems while doing this project.

Secondly, I would like to take an opportunity to show special gratitude to my supervisor, Ts. Dr Muhammad Noorazlan Shah Bin Zainudin, who provide guidance to me in completing this project and always provided me with suggestions, knowledge, and encouragement to improve this project.

Thirdly, do not forget to thank my parents and family for continuously providing enormous support to finish the project on time. They also always help me with what have to do for this project, especially in financial matters.

Finally, thanks to my friends and classmates who always come up with new ideas when I'm having trouble. They are also willing to help me improve my project in many ways.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS AND ABBREVIATIONS

Adam	:	Adaptive Moment
AI	:	Artificial Intelligence
ANN	:	Artificial Neural Network
CNN	:	Convolutional Neural Network
DL	:	Deep Learning
FC	:	Fully Connected
GUI	:	Graphical User Interfaces
IR	:	Industrial Revolution
ML	:	Machine Learning
NLP	:	Natural Language Processing
RNN	:	Recurrent Neural Network
ReLU	:	Rectified Linear Unit Function
ResNet	:	Residual Network
Sgdm	:	Stochastic Gradient Descent with Moment
SVM	:	Suppose Vector Machine
2D	:	2-Dimensional
3D	:	3-Dimensional

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Project Overview

This study aims are to classify the chili fruit grading using based on the size and its variety of chili fruit from its colour in agriculture industries. Agriculture is considered as the most important economic and social sectors for humans [1]. Worldwide, agriculture is a $5 trillion industry for its importance of providing food, raw materials and also employment opportunities within the social community. The classifying agricultural products need necessary variables such as the color and size must be considered. Then, this study aims to design and implement a deep learning model using Convolution Neural Network (CNN) in terms of recognition accuracy using ADAM and SGDM optimizer with various architecture. This study also measures the size of chili fruit according to its length from the calyx and apex to classify the chili fruit grading that need to be classified as unmature, moderate mature and mature. This work starts collecting the number of samples that have been collected from the farms. Then, it will categorize the images depending on their size and colors. After classifying the size and color, the analysis needs to be conducted to measure the accuracy of the classification. In this work, the CNN with a three-layer operation is used, consisting of a convolution layer, a maximum pooling layer, and a fully connected layer.

1.2 Objective

There are several objectives for completing this work, as follows: -

1) To determine the size of chili by measuring the length based on its calyx and apex using geometric coordinate.
2) To analyze the maturity category of chili using ADAM and SGDM optimizer with various Convolution Neural Network (CNN) architecture in terms of recognition accuracy.
3) To classify and analyze the class varieties of chili according to its color using ADAM and SGDM optimizer with various Convolution Neural Network (CNN) architecture in terms of recognition accuracy

1.3 Problem Statement

Nowadays, the agricultural sector is considered as important matters to the global economic system as its growth could affect economic development and activities. Thus, it certainly has made some significant contributions throughout the developing countries in expanding their national economy [2]. As the economy grows, so did agriculture. With the shifting in procedures and techniques in agriculture, the traditional harvesting process are unable to recognize the subject as good as a human does. A slow process leads to an inaccurate and inefficient result, while increasing the production cost. Therefore, the detection and classification process is differs than the previous approach.

By using traditional approach, it's difficult for farmer or manpower to detect the maturity and class variety of chili fruit. Small mistakes or missteps are usually unavoidable when using manpower, especially during estimating the maturity and class variety of the chili fruit because human eyes are prone to errors and tends to inaccurate. This approach represents an innovative, feasible, and economical alternative for farmers who are require the accurate size and grading of chili for daily maturity estimation. In consequences, this work also aims to solve the problem by replacing manpower using Artificial Intelligence (AI) towards the Industrial Revolution (IR 4.0) by using deep learning method such as CNN.

In addition, when analyzing the pixel size of an image, it tends to a difficult situation where the classification of 2D images need to be converted into 1D vector. It tends to lose the characteristics as spatial features of the picture space when the first generation of Artificial Neural Network (ANN) is used. The arrangement of pixels in an image into vector files containing locations or spatial information is referred to as spatial features. In such cases, the image is not associated with any data. Spatial features frequently include a variety of natural or man-made borders or shapes that aid in visualizing spatial data and manipulating networks [3]. For example, if the algorithm is attempting to differentiate between dogs and cats, the snout width and ear length must be specified explicitly as data points. However, by utilizing CNN, these spatial elements are recovered from the image automatically.

Thousands of parts must be extracted to make CNN more optimal. CNN could recognize important elements making it a good solution for computer vision and images classifications.

1.4 Scope of Work

This project aims to analyze the maturity category and class varieties of chili fruit using various architecture of CNN. The length of chili fruit is being measured based on two points from its clayx and apex using geometric coordinate using distance tool in MATLAB with superimposed interactive line. Data collection uses to perform an analysis to classify the maturity and class variety of chili fruit with Graphical User Interface (GUI). Total of 1200 images of the collected in this work. MATLAB Software uses with Deep Learning Toolbox. This toolbox uses to visualize layer activations and graphically monitor training progress. This toolbox is also having the ability to integrate with TensorFlow and PyTorch through the ONNX format and importing models from TensorFlow-Keras and Caffe. The toolbox supports the CNN architecture, which are AlexNet, ResNet-50, Inception Resnet, SqueezeNet and many other pre-trained models.

1.5 Project Significance

Agriculture plays a vital role in most country as foreign trade and provides a living for about 75% of the world's population [4]. Due to the increasing the population, there is a need to raise agricultural output. The main outcome of this project is to be grading, classify and implement a deep learning model to get different maturity categories and varieties of chili fruit using the method CNN in terms of recognition accuracy.

This system would be advantageous for farmers when the proposed grading and classification system is implemented in real world situation. It gives benefit for agricultural industries by exploring new ways to carried out an accurate and efficient chili fruit detection and classification. This innovative method is believed to boost crop yield while reducing costs and maximizing the use of available resources. This contributes to the rising use of digital technology in agriculture, which assist farmers for making better decisions and increasing yields.

Robots for harvesting systems nowadays have been used gasoline which causes air pollution because gasoline is a toxic and highly flammable liquid. Vapors released when gasoline evaporates and materials produced when gasoline is burned such as carbon dioxide, carbon monoxide, nitrogen oxides, particulate matter and noncombustible hydrocarbons that lead to air pollution. So, this work could save the environment by reducing the air pollution. This work also believed provides a significant impact in agriculture in Malaysia and around the world.

1.6 Thesis Outline

The first chapter provides an overview and explanation of a method for analyzing the size of chili fruits for maturity estimation using AI. The research issue, hypothesis, or assumption is also be discussed this chapter. The project aims to classify, analyze, and implement a deep learning model to obtain different variation measures and color of chili for maturity estimation using CNN. The scope of work is also being explained in detail in this part.

The second chapter discusses the literature, including the summary of papers and journals used as a resource for this work. Related topics are being collected and discussed to achieve the project's objectives to solve the problems stated. This chapter also covers the history of AI, methods used in the project, deep learning and related journals.

This third chapter explains how the project is planned and all the requirements needed to meet the project's goals. Each step of the methodology flow chart is broken down into several phases where the detail of each phase is described in detail.

The fourth chapter explains the results and discussion. This section shows the results and analysis of detection and recognition of chili maturity. The use of the methods, algorithm and parameters have also been discussed in this section.

The fifth chapter concludes the result and analysis obtained from the experimental work done in the previous chapters. This chapter also discusses the potential of propose future work for improving the current limitations of this work.

CHAPTER 2

BACKGROUND STUDY

This chapter discusses the literature of several related journals, articles, and previous studies on Artificial Intelligence (AI), Deep Learning (DL) and Convolutional Neural Network (CNN). The previous work related to solving the related problems using Machine Learning, Deep Learning has also been discussed in this chapter.

2.1 Introduction of Artificial Intelligence

Artificial intelligence (AI) is a field of computer science concerned with the creation of intelligent machines capable to work and responding in the same way as humans [5]. One of AI aims is to create algorithm into a machine that can communicate with humans and also able to understand as human intellect, as well as the influence of events and the world on people emotions [6]. In 1950, Alan Turing proposed that a machine's intelligence could be determined by its ability to exhibit intelligent behaviour that is indistinguishable from an intelligent human [7]. In the public sector, artificial intelligence tries to mimic human problem-solving techniques in order to produce more efficient results. The incorporation of AI technology into a computer application with human-computer interaction and data interaction, also known as AI application, is designed to improve performance by replicating human thinking, learning, and problem-solving capabilities [8].

Today, AI technology is widely used in various sectors, with significant impacts on healthcare, industry and agriculture. The world population in agriculture is one of the most extreme industrial sectors because in 30.7% is directly working on 2781 million hectares of agricultural land [9]. The systems in agriculture sector uses AI performs better in terms of accuracy and resilience. Agriculture is a dynamic domain in which it is impossible to generalise situations in order to propose a conventional response.

2.2 Fundamental of Machine Learning

Machine learning is a technique of computer science that evolved from pattern recognition in artificial intelligence and computational learning theory. The analysis and implement of algorithms that learn from data and generate predictions is referred to as machine learning. It is utilized in a variety of computational tasks for which explicit algorithms cannot be designed or coded. While data mining and machine learning are sometimes confused, data mining is primarily concerned with exploratory data processing [10]. Intelligence is the basic distinction between humans and machines when it comes to doing their jobs. The neural system transmits the data it collects to the human brain for perception and action. The data is then organized and recalled through comparison to previously recorded experiences in the brain, and is finally processed during the perception phase [11].

2.2.1 Machine Learning Category

Machine learning aims to develop algorithms which enable computers to learn. Machine learning have a way of identifying statistical trends or other data patterns. Generally, machine learning algorithms can be classified into three categories; supervised, unsupervised and reinforcement in general as shown in Figure 2.1.

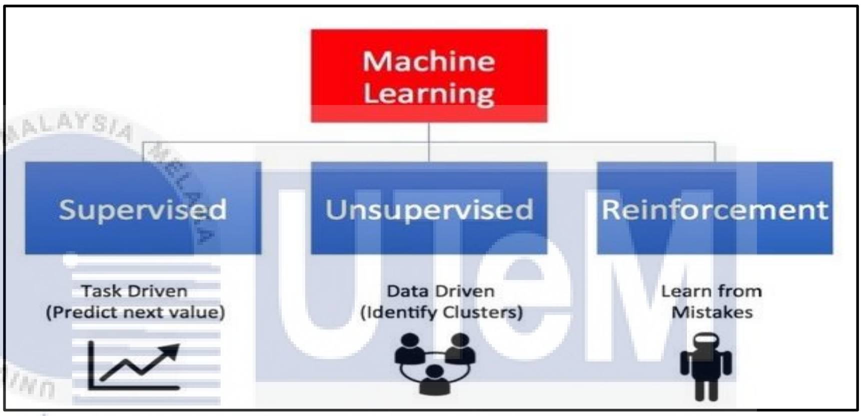


Figure 2.1: Categories of Machine Learning [57]

2.2.1.1 Supervised Learning

The two different types of supervised learning are classification and regression. Regression is used to study the relationship between dependent and independent variables, whereas classification uses an algorithm to accurately assign test findings to specified categories [12]. When it comes to supervised learning issues, there are a number of measured inputs that influence one or more outputs. Input variables might be qualitative, quantitative, or a combination of both. Prediction approaches will be used depending on the differentiation between input variables [13]. Categorical, discrete, and factor variables are all examples of qualitative variables.

The goal of supervised learning is to develop an artificial system that can learn to map inputs to outputs and predict the system's output in the new inputs [14].

2.2.1.2 Unsupervised Learning

In 1970, Marr pioneered unsupervised learning regarding the neocortex model. Geoffrey Hinton and Terrence Sejnowski introduce a model of learning called the Boltzmann machine in 1986 [15]. Unsupervised learning is the second machine learning technique, in which unlabeled data is used to train the algorithm that it is employed on data with no previous labels. The goal is to look into the data and see if there is any new structure there. The programme works out the data and clusters technique with new labels based on the data segments [16].

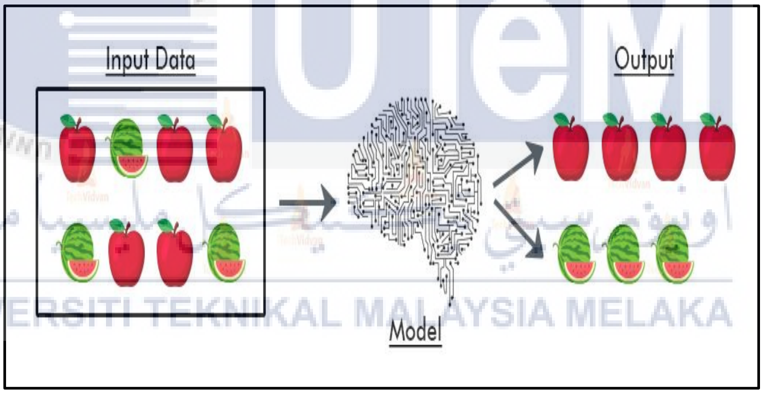


Figure 2.2: Unsupervised Learning Technique [58]

As shown in Figure 2.2, mixed fruit data for the model is used to distinguish the type of fruits. This method will classify objects based on their shape, size, and color and it will learn and differentiate between each of them. A cluster is formed by data points that are similar to those of the same fruit. It trains the model to immediately learn about the input and operate on it. The data has already been solved, the model may then classify data in distinct categories once it is clustered and categorised. Cluster analysis is a sort of machine learning that divides data into clusters that are unlabelled, unclassified, or uncategorized. Rather than reacting to input, cluster analysis finds commonalities in the input and reacts to the presence or absence of those commonalities in each case. This new method of information aids in the detection of out-of-place data items [17].

2.2.1.3 Reinforcement Learning

A reinforcement learning-based machine learning approach aids in evaluating which action provides the most substantial benefit over time. Since learning decisions in the Reinforcement learning approach are dependent, this method relies on interacting with the environment. Dependent decisions should be labelled, and it also aids and improves AI, which requires a great deal of human interaction [18].

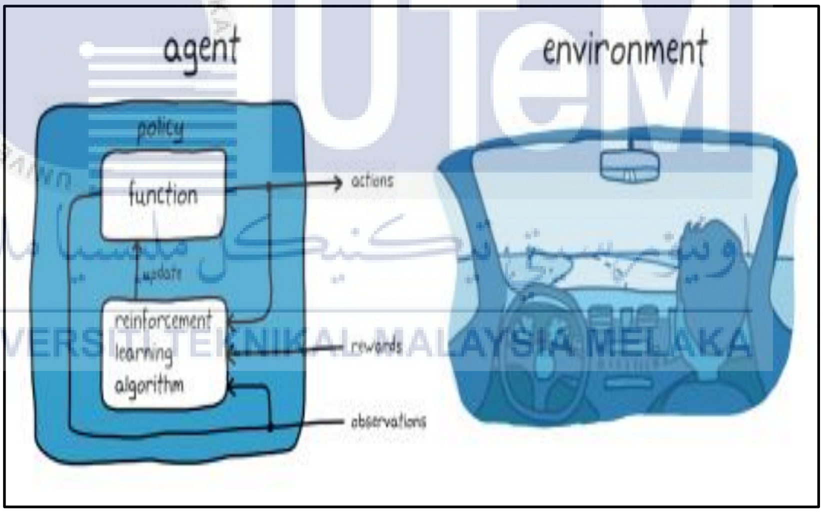


Figure 2.3: Reinforcement learning in autonomous parking [59]

The learning of the road map by using the car in the autonomous parking example is handled by a training algorithm, as shown in Figure 2.3. Based on the sensor readings, actions, and incentives collected, the trained model is in charge of fine-tuning the agent policy. After the training, the vehicle's computer should park utilising solely the tuned policy and sensor readings.

2.3 Fundamental Deep Learning

Since 1979, scientists have studied deep neural networks. However, when it was employed for unsupervised learning to reduce data dimensions in 2006, it formed a new machine learning research subject [19]. Deep learning is a term used to describe a series of machine learning algorithms that employ numerous layers of nonlinear processing units [20]. It enables a programme to build more complex ideas from basic ones. Including more fundamental ideas like corners and contours, which are represented in terms of edges, a deep learning system can reflect the idea of a person's picture [21].

The structure of the human brain inspires the architecture of the neural network as deep learning as shown in Figure 2.4. Neural networks are programmed to discover patterns and classify diverse sorts of knowledge in the same way as human brains do. Individual layers of neural networks can be viewed as a type of filter that works from the most obvious to the most subtle, increasing the possibility of correctly recognising and providing the desired output. [22].

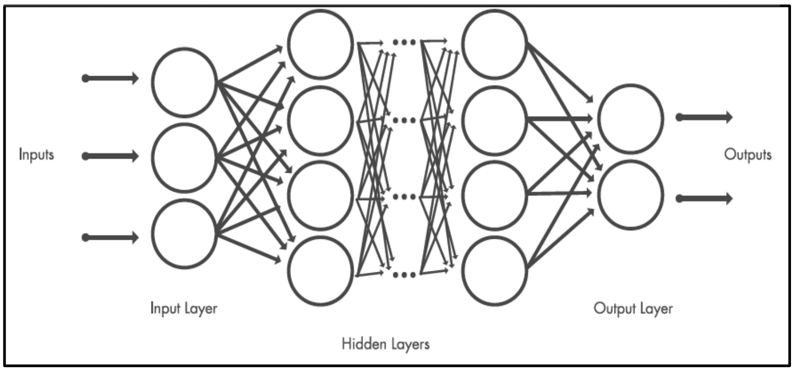


Figure 2.4: Illustration of a Deep Learning Model [60]

2.3.1 Deep Learning Category

Deep learning is a term that refers to variety of type for learning from data. Neural networks have a hierarchical architecture that can be used to solve number of problems. Therefore, deep learning can be classified into two-type in general, as shown in Figure 2.5.

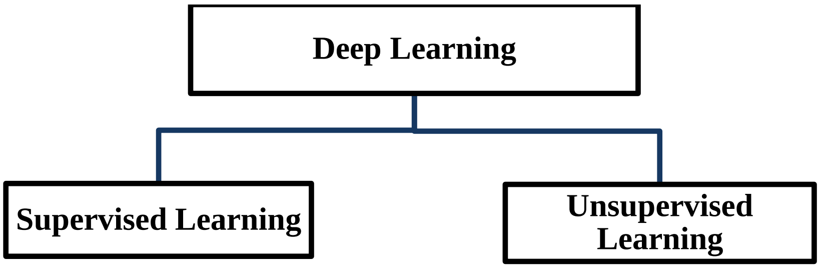


Figure 2.5: Type of Deep Learning

2.3.1.1 Supervised Learning

It splits two types of problem for pattern classification and regression in supervised learning. To begin, the model uses the labels provided with the data to learn the differences between data objects belonging to different classes. In both classification and regression tasks, the model learns to relate input to expected output. ANN, CNN, and Recurrent Neural Network(RNN) are examples of supervised learning methods [23].

An Artificial Neural Network (ANN) is a component of a computing system that is designed to evaluate and make decisions in the same way as the human brain does. ANN is a deep learning building block that can handle tasks that humans find impossible or extremely difficult. ANN work in the same way as the human brain does. Artificial Neural Networks have three layers: an input layer, a hidden layer and output layer. Each layer is built up of neurons, which are interconnected adaptive processing units [21]. The cell body of each neuron in the human brain is responsible for calculating information by forwarding information to hidden neurons and giving final output. Based on the inputs given to the input layer during the training phase, the ANN learns to detect patterns [24].

CNN are AI systems that use multi-layer neural networks to learn important properties from images and perform tasks which are object classification detection, and segmentation [25]. The CNN method has four types of layers which is Convolutional Layer, Activation Layer, Max Pooling Layer, and Fully Connected Layer. CNN main benefit over its predecessors is that can detect crucial qualities without the need for human interaction. This is why CNN would be an excellent solution for picture categorization and computer vision challenges [26].

RNN is a type of artificial neural network that works with sequential or time series data. Deep learning approaches face issues in language translation, Natural Language Processing (NLP), speech recognition, and image captioning, to name a few. This approach is used in popular application like Siri, Google Translate, and voice search [27].

2.3.1.2 Unsupervised Learning

Unsupervised learning occurs when there is simply input data and no corresponding output to map. Though algorithms might be able to reveal the data's intriguing structure. In clustering challenges, unsupervised learning algorithms such as the k-means method are used [22]. In exploratory data analysis and data mining, K-means clustering is extremely useful. Big data sets have become more common as computer power has expanded because of its ease of implementation, computational efficiency, and low memory utilisation, k-means clustering has remained popular in comparison to competing clustering algorithms. [30].

2.4 Convolution Neural Network (CNN)

The foundation of the convolutional neural network started from the discovery of Hubel and Wiesel in 1959 [31]. Among the numerous deep learning architectures, CNN is a specific sort of multilayer neural network for spatial data.

CNN architecture is inspired by the visual perception of living beings. CNN is mostly used for image recognition and has a strong feature extraction capability [32]. The various CNN architectures including LeNet, AlexNet, VGGNet, GoogleNet, ResNet and ZFNet [33].

CNN are a type of neural network that is particularly well adapted to the intelligent processing of visual data. Convolution layers, pooling layers and fully linked layers are included in this multilayer neural network architectural variation [34] as shown in Figure 2.6.

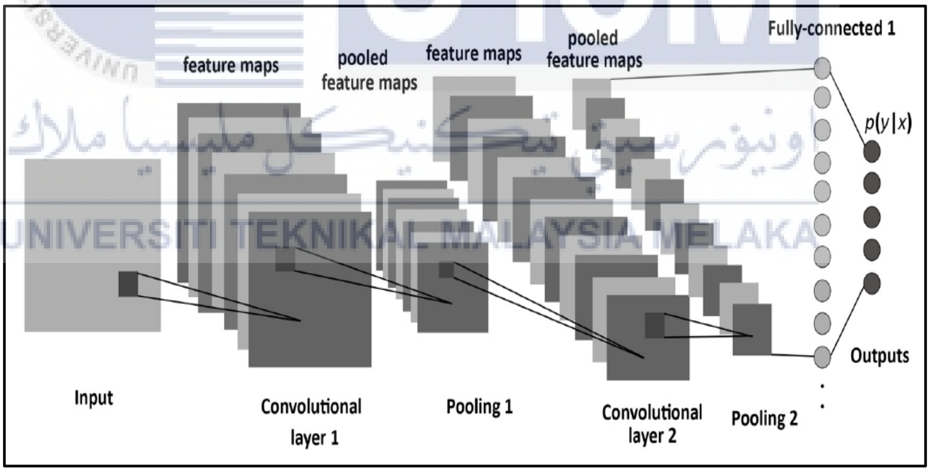


Figure 2.6: Operation of Convolution Neural Network (CNN) [61]

2.5 MATLAB Deep Learning Toolbox

Deep Learning Toolbox is a platform for creating and deploying deep neural networks, including algorithms and pre-trained models [35]. The toolbox can classify and regression on image, time-series, and text data using convolutional neural networks and extended short-term memory networks. It also able to build a network architecture such as generative adversarial networks using automatic differentiation, custom training loops, and shared weights. Besides, the Deep Network Designer app uses graphics to construct, evaluate, and train networks.

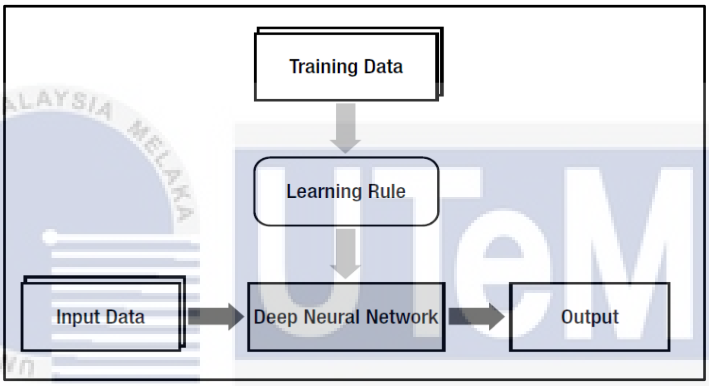


Figure 2.7: Concept of Neural Network [36]

The deep neural network is two or more hidden layers make up a multi-layer neural network. Although it may appear to be very simplistic, this is the fundamental essence of DL. [36]. Figure 2.7 illustrates the concept of Deep Neural Network and its relationship to machine learning.

2.6 Table of Comparison

In Table 2.1 shows a number of previous related work done by several researchers. Various parameters such as method used and the image categorization and also the analysis of model's performance in terms of recognition accuracy also be reported.

Table 2.1: Table of Comparison

No.	***Author***	***Journal***	***Method***	***Accuracy***	***Criteria Feature***
1.	Shiv Ram Dubey and Anand Singh Jalal	Fruit and vegetable recognition by fusing colour and texture features of the image using ML [37]	ML	88.7 %	Color Texture Feature
2.	T. Ishikawa, A.Hayashi,S. Nagamatsu,Y .Kyutokn, I. Dan,T.Wada, K. Oku, Y.Saeki, T. Uto	Classification Of Strawberry Fruit Shape By ML [38]	ML	70 %	Shape

3.	M. Atas_, Y.Yardimci, A. Temizel	A new approach to aflatoxin detection in chili pepper by machine vision [39]	ML	83.26 %	Feature Extraction
4.	O. Cruz-Domínguez , J.L. Carrera-Escobedo, C.H. Ortiz-Rivera,	A novel method for dried chilli pepper classification using Artificial intelligence [40]	ANN	82.7 %	Size Color
5.	Mandeep Kaur1, Reecha Sharma2	Quality Detection of Fruits by Using ANN Technique [41]	ANN	80 %	Shape Color Size
6.	A Taofik, N Ismail, Y A Gerhana, K Komarujaman and M A Ramdhani	Design of Smart System to Detect Ripeness of Tomato and Chili with New Approach in Data Acquisition [42]	K-Means Clustering	80 %	Color

7.	S.Deepika, Dr.R.Punidha	Fruit Maturity And Disease Detection Using Artificial Neural Network[43]	ANN	83 %	Size Shape
8.	Norasyikin Fadilah,Junita Mohamad Saleh, Haidi Ibrahim,Zaini Abdul Halim	Oil Palm Fresh Fruit Bunch Ripeness Classification Using Artificial Neural Network [44]	ANN	86.67 %	Color
9.	Kazi Abu Taher,Billal Mohammed Yasin Jisan, Md. Mahbubur Rahman	Network Intrusion Detection using Supervised Machine Learning Technique with Feature Selection [45]	SVM	82.34 %	Feature Selection

2.7 CNN Architectures

The CNN architecture is a multifunctional extractor which could classify the images, identify the objects, and divide the images as well as other complex tasks.

There are few numbers of well-known CNN architectures used, namely AlexNet, Inception ResNet and ResNet-50.

2.7.1 AlexNet

The AlexNet architecture as shown in Figure 2.8 is more complex, larger and has layers stacked on top of layers. This architecture was the first to popularize the Convolution Network in Computer Vision; it won the ILSVRC ImageNet competition in 2012 and had an error rate among the top 5 of 15.4 %, compared to 26.2 % for the next lowest network [46]. AlexNet Architecture is also known as a wide and deep convolution neural network with 25 layers that is in convolution, maximum pooling, drop out and fully connected. After each convolution, there is a ReLU activation function, and the layers are fully connected. If the layer is dropped, there is a 50% chance that it will be dropped and determine high-level features, the first layer operates as a feature extractor [47].

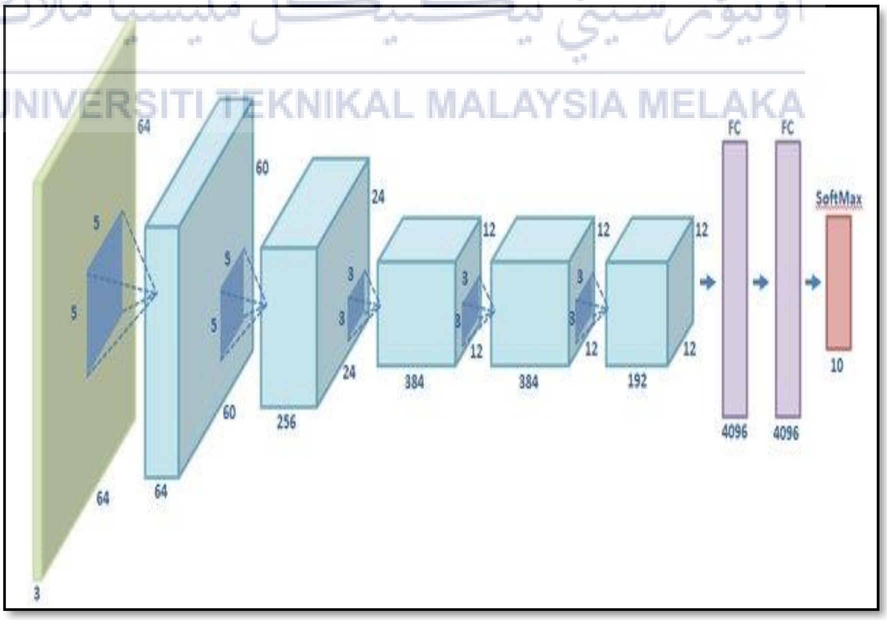


Figure 2.8: AlexNet Architecture[62]

2.7.2 Inception Resnet

The Inception-Resnet-v2 structure is a hybrid of the Inception and Residual Network (ResNet) structures [48]. Inception-ResNet-v2 is a convoluted neural network in 164 layers that have been trained with over a million datasets. It is built on the foundation of Inception structure and Residual connection with multi-size convolution layer mixed with residual connection in Inception-Resnet block. The introduction of waste extensions not only addresses the issue of deterioration caused by internal structures but also reduces training time by half.

2.7.3 Residual Network 50 (ResNet-50)

The ResNet architecture is based on deep designs that have demonstrated excellent convergence and accuracy [49]. ResNet was created using numerous stacked residual units and a variety of layer counts: 18, 34, 50, 101, 152, and 1202. The number of operations, on the other hand, can vary depending on the architecture and for ResNet-50 is composed of 49 convolution layers and the layer clings fully at the ends[49]. The ResNet will learn increasingly complicated features as time goes forward. This architecture will recognise images with the first layer learning to detect edges, the second layer learning to recognise textures, and the third layer learning to find objects[49].

CHAPTER 3

METHODOLOGY

This chapter covers how the project's objectives are placed into action and each of the project's strategies. This chapter also includes a flow chart explaining and illustrating the project's process in depth. The software, method and block diagram are also explained in this chapter.

3.1 Research Methodology Flow Chart

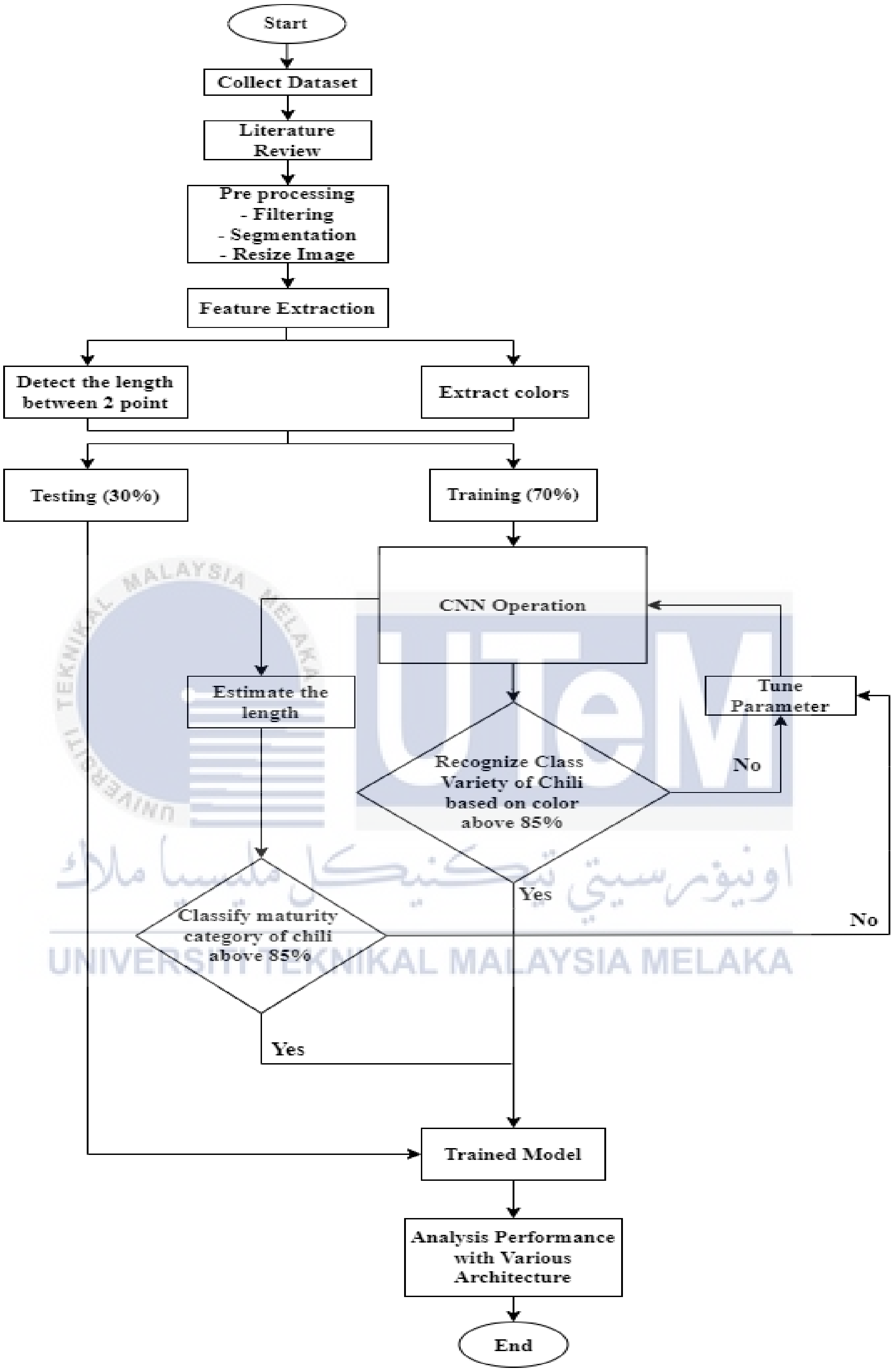


Figure 3.1: Project Flow Chart

This research begins with a literature review on Artificial Intelligence, Machine Learning and Deep Learning at the beginning of the flow. Many journals and articles were consulted to understand better this project's relevant approaches, techniques, and methodology.

The next stage is to collect the image of chili dataset. The chili dataset is considered to be challenging to be collected due to some conditions. Hence, the first step is obtaining the image by taking the photo for each chili according to its size according to its maturity category and the varieties class based on its color. Then, all the datasets are undergone the preprocessing stages including filtering, segmentation and resizing the image. The noise or unwanted features is removed before pursuing any further process. The technique of separating one image into many pieces is known as picture segmentation. It is used in digital photos to identify the items and pertinent information. Filtering is a method of altering or improving the representation of image. For example, filter is applied to an image to highlight certain elements while removing others [50]. The process of segmenting a digital image into several segments, or groups of pixels, and known as image segmentation. The purpose of segmentation is to make an more image more easier to understand by simplifying or changing its representation [51]. Then, the image pixel is resized from 4160x3120 to 256x256 in this work.

Next, the first objective is to determine the size of chili by measuring the length based on its calyx and apex using distance tool in MATLAB Software. So, the next step is to proceed with the feature extraction as dimension reduction, which dividing and reducing large datasets into smaller groups. The process is easier, and the important characteristic of these large data sets is generated and produced a large number of variables. All these variables require a lot of computing resources to be processed. As a result, feature extraction aids for extracting the best feature from large data sets by selecting and combining variables into set of features at the same time reducing the size of data. These features is considered as an accurate and uniquely describing the real dataset. The chili fruit is measured and divided into its categories by the folder based on its sizes.

The second objective is to analyze the maturity category of chili. The various size of the chili is used to determine the maturity category of chili fruit. The process of classifying and analyzing the maturity category of chili fruit according to its size using CNN. Each input image is filtered through a succession of convolution layers. The Pooling Layers, Activation Layer, Fully Connected Layers (FC) and Softmax Function is applied in this case.

Finally, the third objective is to classify and analyze the class varieties of chili fruit according to its color using CNN. The feature of the image is also could be used to recognize the category based on its color. Coding and simulation need to be done repeatedly for pursuing the process of classifying the maturity category and varieties class of the chili fruit. The coding using the CNN model need to be run in two phases of subsets: training and testing. The CNN method learns the data by defining the characteristics according to its maturity category and the class varieties. In this work, the desired accuracy is defined above 85%, it can train the model and perform the analysis performance with various architecture. However, if the accuracy is under 85%, parameter tuning is performing until the desired accuracy is obtained.

3.2 Software Requirement

The software used in this work is shown in Table 3.1: -

Table 3.1: Software version

No.	Software	Version
1.	MATLAB	R2021a

3.3 MATLAB Software

In this work, the deep learning toolbox with MATLAB installed for the conducting all the required processes.

3.3.1 Deep Learning Toolbox

The Deep Learning Toolbox is a framework to develop and deploy a deep neural network with various architecture. CNN is commonly used to classify and predict various types of data such as images, time series and text. The Deep Network Designer graphically in Deep Learning Toolbox allows you to construct, evaluate, and train networks graphically.

The Experiment Manager tool used in managing multiple deep learning experiments, track training parameters, analyse outcomes, and comparing the code during the experiments by showing the activations layer and training progress. Then, by using the ONNX format, it also enables to interchange models with TensorFlow and PyTorch and importing the models from TensorFlow-Keras and Caffe. The toolbox also numbers of transfer learning such AlexNet, ResNet-50, and Inception Resnet.

.

3.4 System Overview

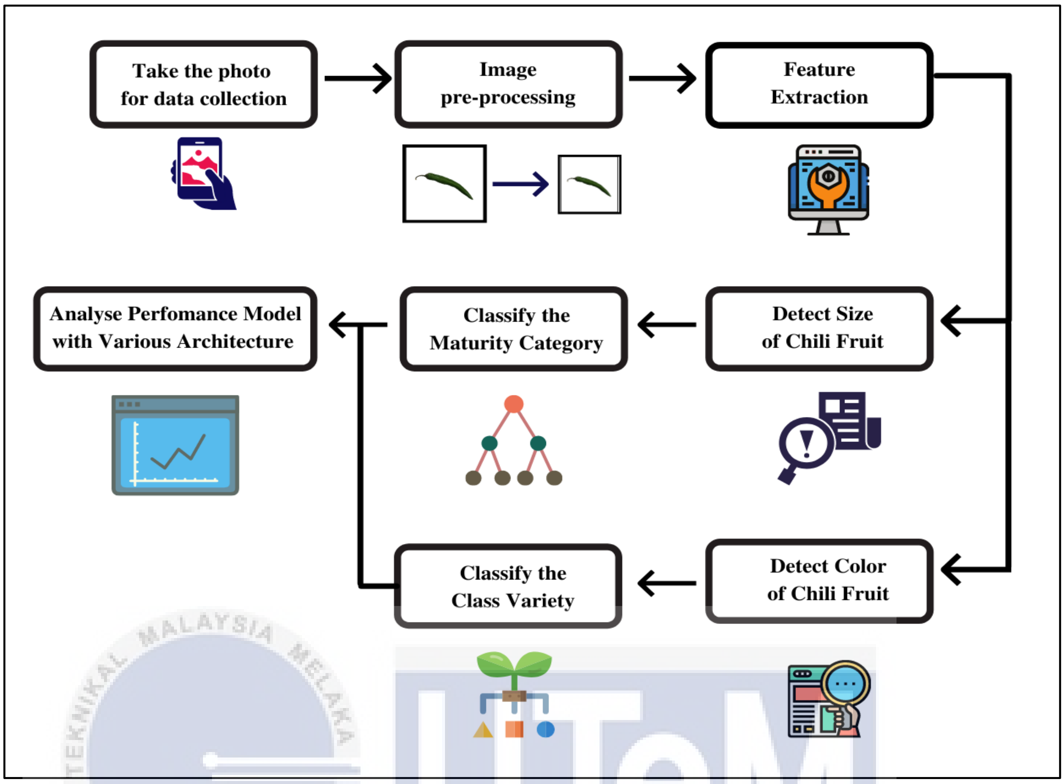

Take the photo for data collection
Image pre-processing
Feature Extraction
Analyse Perfomance Model with Various Architecture
Classify the Maturity Category
Detect Size of Chili Fruit
Classify the Class Variety
Detect Color of Chili Fruit

Figure 3.2: Block Diagram Project

Figure 3.2 shows the block diagram of the entire system overview. This work uses a deep learning model using CNN. Firstly, the image of chili fruit is collected for both categories: maturity and class variety. Image pre-processing is conducted to resize the size to make the process classification in MATLAB software easier. Feature extraction is used to extract the features by measuring the length and to recognize the chili class based on its color. CNN is used as an algorithm to detect the size and color to classify the maturity category and class variety of chili fruit respectively. Lastly, the performance in terms of recognition accuracy is analyzed and compared with various architecture.

3.4.1 Dataset of Chili Fruit

3.4.1.1 Maturity Category of Chili Fruit

There are consisting total of 600 images where 420 images used for training and 180 images for testing. All images are categorized into three maturity categories according to its size; unmatured, moderately matured, and matured. The image is made of frames which the position has been rotated. Table 3.2 shows the sample of instances of each class to represent the category size. The size of each image is set to be 256 by 256 pixels.

Table 3.2: The Maturity Category of Chili Fruit

No.	Maturity Category	Size	Training	Testing
1.	Unmatured	<6 cm	140	60
2.	Moderately Matured	7-11 cm	140	60
3.	Matured	>12 cm	140	60

3.4.1.1 Varieties of Chili Fruit

Total of 70% of images are randomly chosen is used for training, whereas the image for testing is 30%. All the images are divided into three categories based on the color: green chili, red chili, and overriped chili as shown in the sample from Table 3.3

Table 3.3: The Variety Class of Chili Fruit

No.	Variety Class	Color	Training	Testing
1.	Green Chili 	Green	140	60
2.	Red Chili 	Red	140	60
3.	Overriped Chili 	Brown	140	60

3.4.2 Operation of Convolution Neural Network (CNN)

The architecture of CNN is represented as a series of layers that consists of four layers which known as convolution layer, activation layer, max pooling layer and fully connected layer as shown in Figure 3.3.

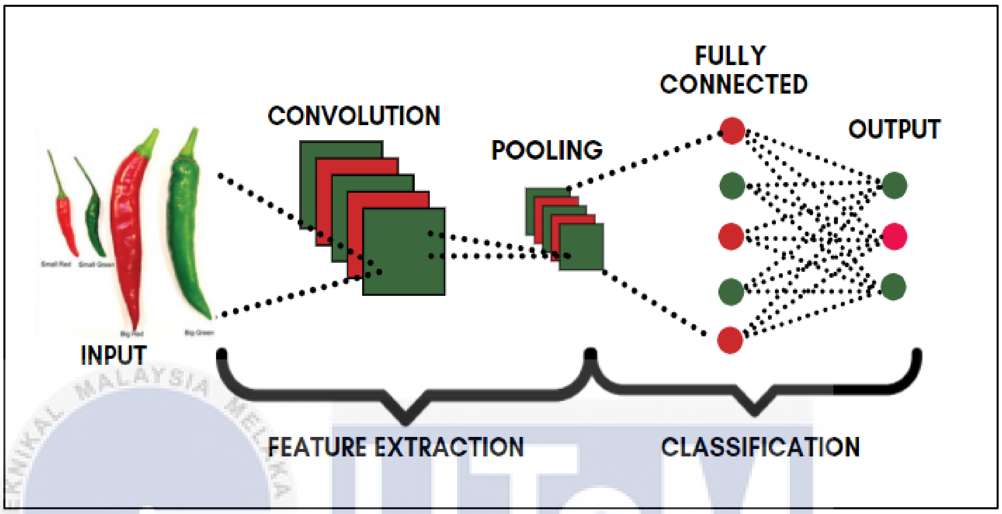


Figure 3.3: Operation layer for CNN

3.4.2.1 Convolutional Layer

The first layer of a CNN is a convolutional layer. It has a kernel to learn and can extract features of different images. Each connection represents a distinct kernel that is utilised in the convolution procedure to construct the output or activation map of the current convolutional neuron.

The convolutional neuron uses a unique kernel and the output of the previous layer's matching neuron to conduct an elementwise dot product as many intermediate results where the distinct kernels is produced. The learnt bias is added to all of the intermediate outcomes to create the convolutional neuron. The operation converts all pixels in its receptive field into a single value. In this layer, the mathematical process of convolution is performed between the input image and a filter of a particular size.

The kernels always have the same depth as the input and its small size. Kernel at neurons is connected to a small region of the input because it is highly inefficient to link all neurons to all previous outputs in the case of inputs of high dimensions such as images. For example, of image size of a 256 x 256 with 65536 pixels and the first layer has 100 neurons, it will result in 6553600 parameters. The full dimension of the input will support the weight from each neuron. For the dimension of the kernel input, a neuron will hold weights.

The kernels slide across the width and height of the input, extracting a high-level feature and produce a 2-dimensional activation map. A kernel slide is provided a parameter at the stride at which it is given a parameter. A convolutional layer output is created by stacking the activation maps that are used to determine the input of the following layer. The size will be lowered even more when more convolutional layers are used. As a result, the size of image is substantially reduced, resulting in information loss and a vanishing gradient problem, as illustrated in Figure 3.4.

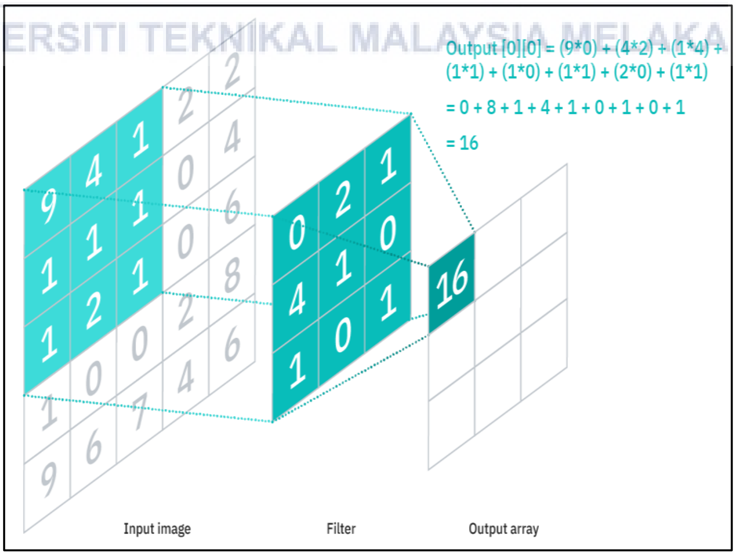

Output [0][0] = (9*0) + (4*2) + (1*4) + (1*1) + (1*0) + (1*1) + (2*0) + (1*1)
= 0 + 8 + 1 + 4 + 1 + 0 + 1 + 0 + 1
= 16
16
Input image
Filter
Output array

Figure 3.4: Operation Convolutional Layer [52]

3.4.2.2 Activation Function

An activation function explains how the weighted sum of input is turned into an output from a node or nodes in a layer of a neural network. The activation function is also known as a "transfer function." When an activation function's output range is restricted, it's referred to as a "squashing function". Most of activation functions are nonlinear, known as "nonlinearity" in the layer or network architecture. The chosen of activation function gives a significant influence on the neural network's capabilities and performance, and different activation functions may be employed in different regions of the model. The hidden layer has the activation function to get input from another layer and outputs to a different layer. There are three activation functions used in hidden layers, which is Rectified Linear Activation (ReLU), Logistic (Sigmoid) and Hyperbolic Tangent (Tanh).

The Rectified Linear Unit Function or ReLU activation function is the most commonly used activation function in hidden layers. The ReLU activation function is implemented and effective at overcoming the limitations of other previously popular activation functions, such as Sigmoid and Tanh. ReLU is a piecewise linear function, when the input is positive it will directly produce an output. Or else, an output becomes zero because a model that uses this activation function is faster to train and frequently produces superior performance.

The logistic function is another name for the sigmoid activation function. The logistic regression in classification also uses the same activation function. The function accepts any real value as an input and returns a value between 0 and 1. When input is greater, the closer the output to 1, and if the input is smaller, the output will close to.

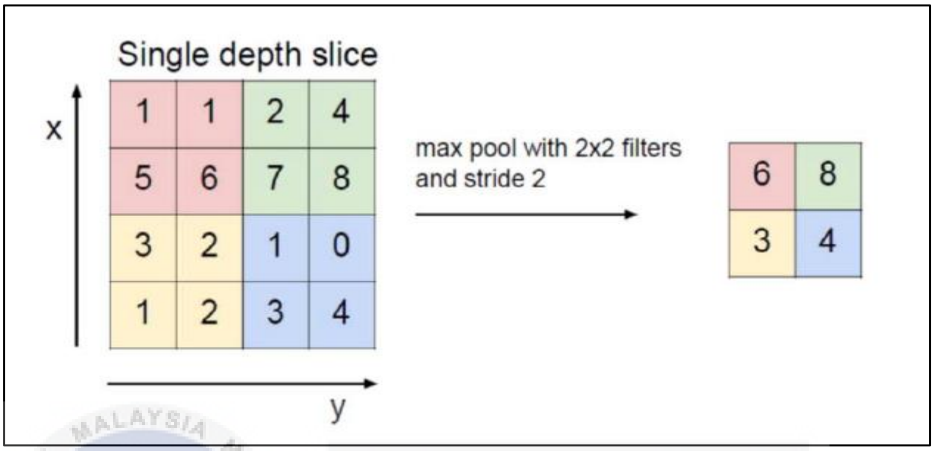
Softmax activation function commonly employed as the final activation function of a neural network to normalise the network's output to probability distribution across anticipated output classes in multinomial logistic regression. The softmax function accepts a vector z of K real values as an input. It converts it to a probability distribution with K probabilities proportional to the input number’s exponentials.

3.4.2.3 Max Pooling Layers

An additional layer called a pooling layer is added after the convolutional layer. The fundamental goal of pooling is to minimise the complexity of subsequent layers by downsampling. In the context of image processing, it's comparable to lowering the resolution. Pooling has no effect on the number of filters. When the images too huge, the pooling layers portion will minimise the number of parameters. Spatial pooling, also known as subsampling or downsampling, reduces the dimensionality of each map while still maintaining important information.

There are a few different kinds of spatial pooling, which are maximum pooling, average pooling and sum pooling. The major goal of this layer is to lower the size of the convolved feature map and the computational expenses associated with it. It is carried out by limiting the connections between layers and operating on each feature map independently. The average pooling is calculated by taking the most significant element, and sum pooling refer to the sum of all elements in a feature map.

This work uses maximum pooling layer, and the function feature is a map pooling method that estimates the maximum or biggest value in each patch. It divides the image into sub-region rectangles and only returns the greatest value of each sub-region. The number 2 x 2 is one of the most popular sizes used in max-pooling. When pooling is done in the top-left 2 x 2 blocks, it advances two and focuses on the top-right portion, as illustrated in Figure 3.5. This indicates that stride two is utilised in the pooling process. Stride 1 can be used to avoid downsampling, which is uncommon.



1	1	2	4
5	6	7	8
3	2	1	0
1	2	3	4

6	8
3	4

Figure 3.5: Max Pooling Layer [53]

3.4.2.4 Fully Connected Layer

The weights, biases, and utilized to connect the neurons between two layers make up the fully connected layer. These layers are frequently placed before the output layer, and they make up the final few levels of the CNN design. The primary responsibility is to extract features from CNN output and connect the feature extraction stages to the Softmax. An ultimately linked layer in a conventional CNN generally consists of 2-3 layers entirely connected. Figure 3.6 shows that the input image from the previous layers is flattened and sent to the fully connected layer. The flattened vector is then passed through a few additional levels before reaching the fully linked layers, which is where the mathematical function operations are normally performed. The classification process gets started at this point. So, activation function such as softmax or sigmoid will classify the outputs.

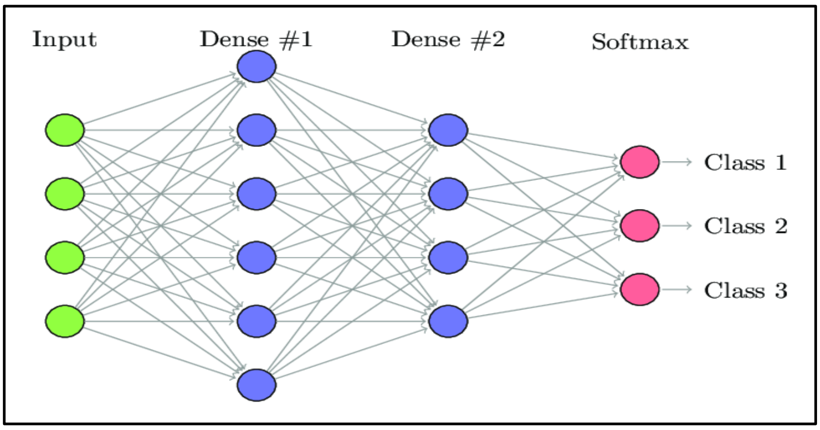


Figure 3.6: Fully Connected Layer [54]

3.4.3 Adaptive Moment Estimation (Adam) Optimizer

Adaptive Moment Estimation (Adam) is a deep learning neural network training technique which uses an adaptive learning rate optimization algorithm. It's a deep learning model training optimization approach that replaces stochastic gradient descent. Adam combines most features of the Stochastic Gradient Descent with momentum and RMSProp methods to provide an optimum solution for sparse gradients. Adam is simple to set up, and the default settings work well for most problems. It uses squared gradients to scale the learning rate, similar to RMSprop, and it takes advantage of momentum provided by the gradients. It's comparable to SGD with momentum in that it's a moving average rather than the gradient itself. Adam is an adaptive learning rate technique that calculates individual learning rates for various parameters. Adam employs estimations of the first and second moments of the gradient to alter the learning rate for each weight of the neural network.

3.4.4 Stochastic Gradient Descent (SGD) with Momentum

The Stochastic Gradient Descent (SGD) with Momentum method aids in the acceleration of gradient vectors in the right direction. These famous gradient descent optimization algorithms also can be used to speed up or to improve deep neural network training. This optimizer eliminates the requirement for momentum hyperparameter calibration, enables a much higher learning rate, accelerates training, and increases final accuracy. SGD iteratively updates the model parameters by shifting the direction of the negative gradient determined on a mini batch scaled by the step length, which is commonly referred to as the learning rate. To achieve convergence, the learning rate must be decayed as the process progresses. For big datasets, batch gradient descent performs redundant calculations because it recomputed gradients for comparable cases before each parameter change. SGD eliminates duplication by completing one update at a time.

CHAPTER 4

RESULTS AND DISCUSSION

This chapter describes the experimental process to determine the chili maturity using the CNN, as well as accuracy recognition results and analysis using various architectures. The experiment on learning rate optimization algorithm used for accuracy recognition also been discussed.

4.1 Results and Analysis of Experiment

The work aims to recognize the maturity category and variety of class for chili fruits. The CNN is utilized to detect the size and color of chili using the parameter listed in Table 4.1.

4.1.1 Measure the size of Chili Fruit

The chili fruit is determined by measuring the length from its calyx and apex. The calyx refers to the part of the stem that links to the chili fruit's top and the apex is the rounded point of the fruit as shown in Figure 4.1.

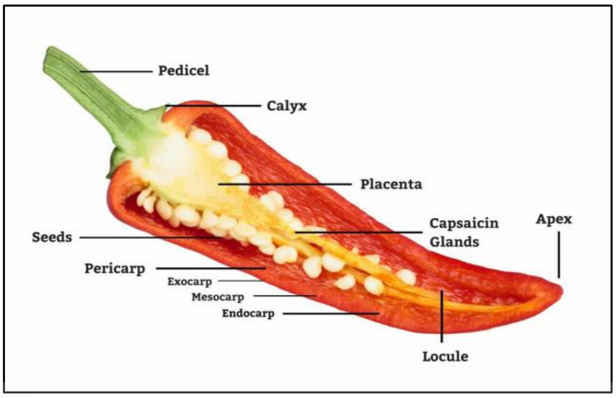


Figure 4.1: The Chili Fruit from Calyx to Apex [63]

Segmentation is used to measure length of the chili fruit from two points where the calyx and apex. Segmented image is converted into a binary image in order to distinguish the background from the objects as shown in Figure 4.3. To measure the length between its calyx and apex, a distance tool is used, which consists of an interactive line superimposed on an image, along with a text label indicating the distance between the line endpoints.

The dataset for maturity category consisting of three classes; the estimation size in less than 6 cm representing unmature, moderate mature in estimation size from 7 cm to 11 cm and mature in estimation size above 12 cm.

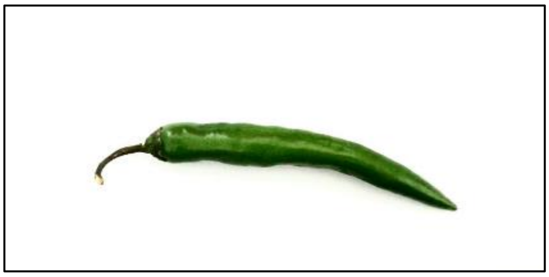


Figure 4.2: Input Image

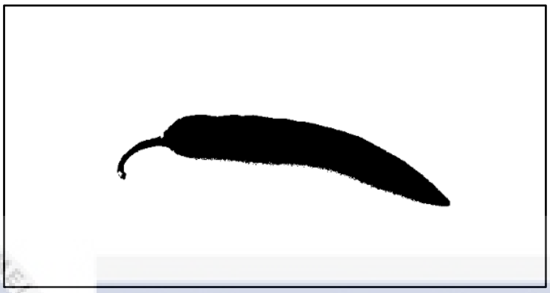


Figure 4.3: Segmentation Image

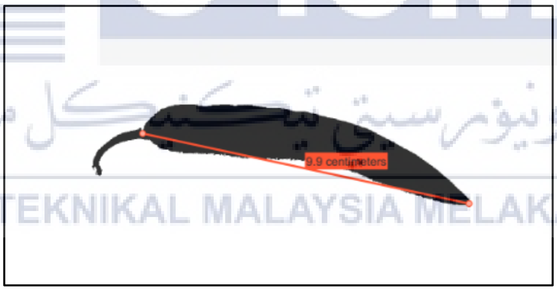


Figure 4.4: Size Chili from Calyx to Apex

4.1.2 Parameter of Experiments

The study employed the CNN approach to analyze 1200 image to determine the maturity category and the chili variety. CNN has two phases: training and testing with 100 epochs used. To classify the training dataset, each sample that has been trained must be able to classify the testing dataset. A slower learning rate may allow the model to learn a more optimal or even globally optimal set of weights but training usually takes much longer. The batch size is the number of data items utilized to train the model. Table 4.1 shows the experiment parameters use using the CNN method.

Table 4.1: Parameter use for training

Parameter	Value
Training Epoch	100
Learning Rate	0.01, 0.001 & 0.0001
Maximum Iteration	400
Batch Size	64

4.1.2.1 Classification using CNN

The three-layer in methodology gives the excellent result for the detection of the maturity category and its variety. Data augmentation have been done before fitting the CNN model as translation invariance. The experiment divided dataset ratio of 70 to 30 for training and testing subset. To evaluate the performance, the experiment used a different optimizer for ADAM and SGDM optimizer.

(a) Experiment on Maturity Category of Chili Fruit

A 2D image of chili fruit was taken from the dataset and used as an input image. After the chili images has been segmentation, it has been classified according to different sizes to the stages of maturation. The chili fruit maturity category received a performance accuracy for recognition of over 85% using CNN which is considered as an outstanding performance in this experiment.

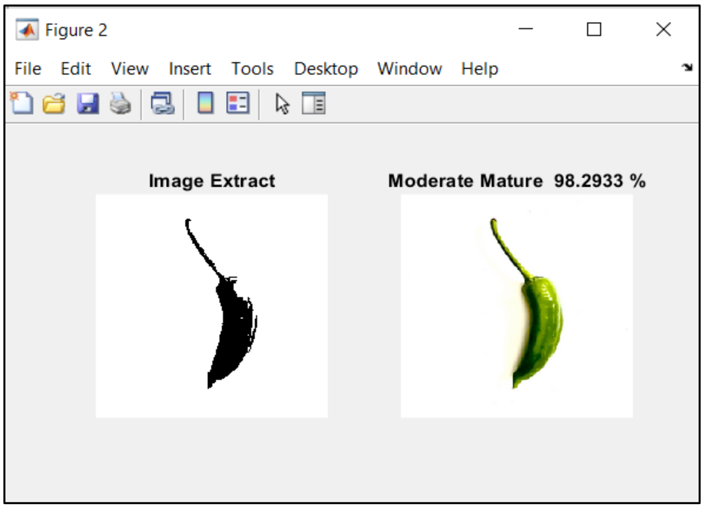


Figure 2
File Edit View Insert Tools Desktop Window Help
Image Extract
Moderate Mature 98.2933 %

Figure 4.5: Performance Accuracy for each Chili Fruit

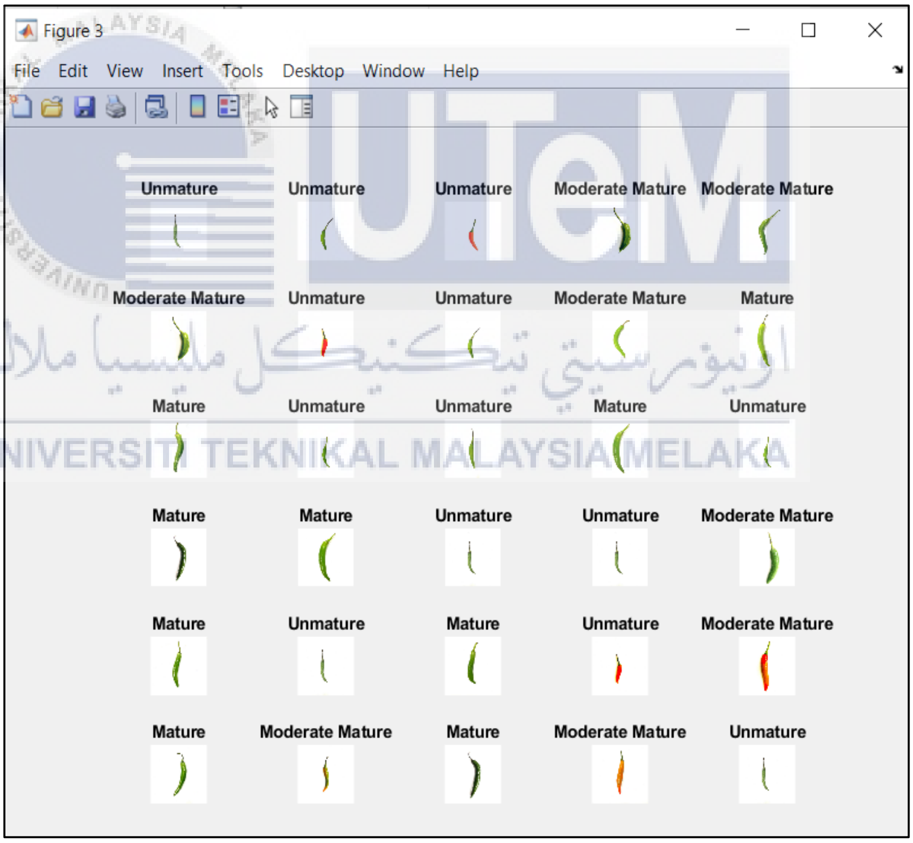


Figure 4.6: Classification for Maturity Category of Chili Fruit

(b) Experiment on the Variety Class of Chili Fruit

After chili fruit is recognized, it has been classified according to its color for the class variety. In this part, the desired accuracy also been defined above 85%. As shown in Figure 4.7, the accuracy is obtained above 93% for detecting the class variety of chili.

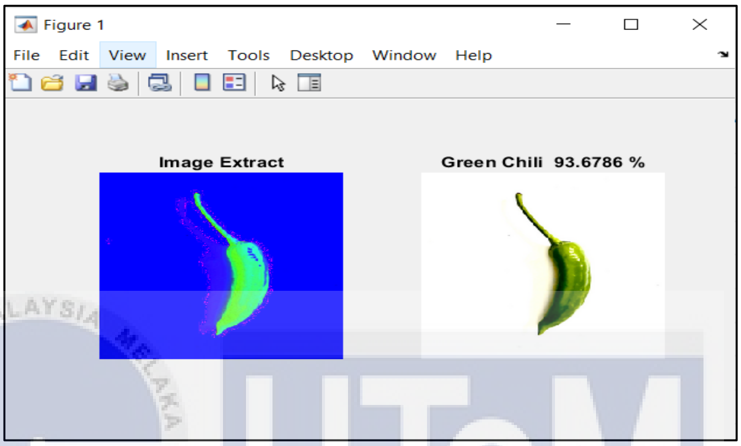
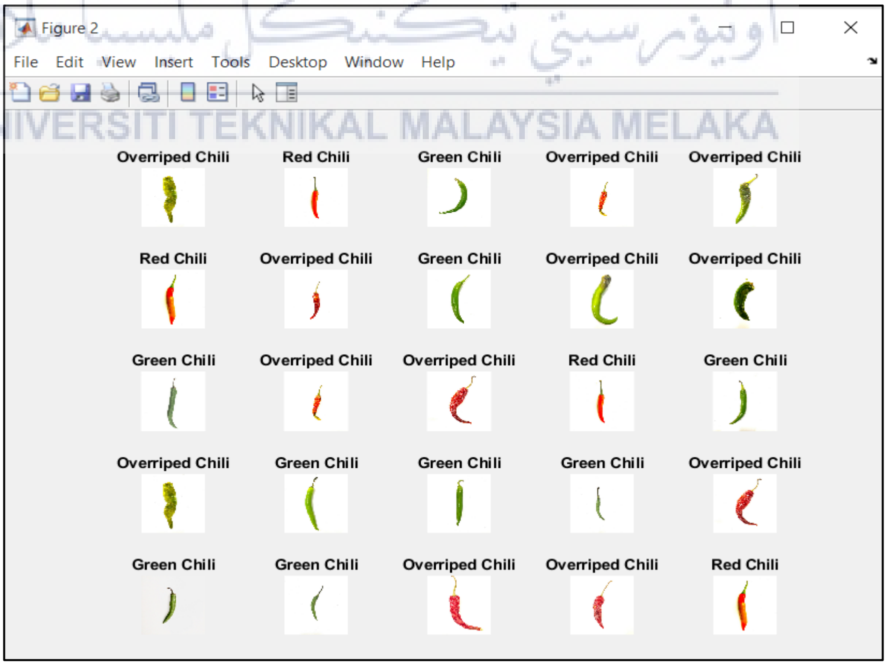


Image Extract

Green Chili 93.6786 %

Figure 4.7: Performance Accuracy for each Chili Fruit



Overriped Chili Red Chili Green Chili Overriped Chili Overriped Chili

Red Chili Overriped Chili Green Chili Overriped Chili Overriped Chili

Green Chili Overriped Chili Overriped Chili Red Chili Green Chili

Overriped Chili Green Chili Green Chili Green Chili Overriped Chili

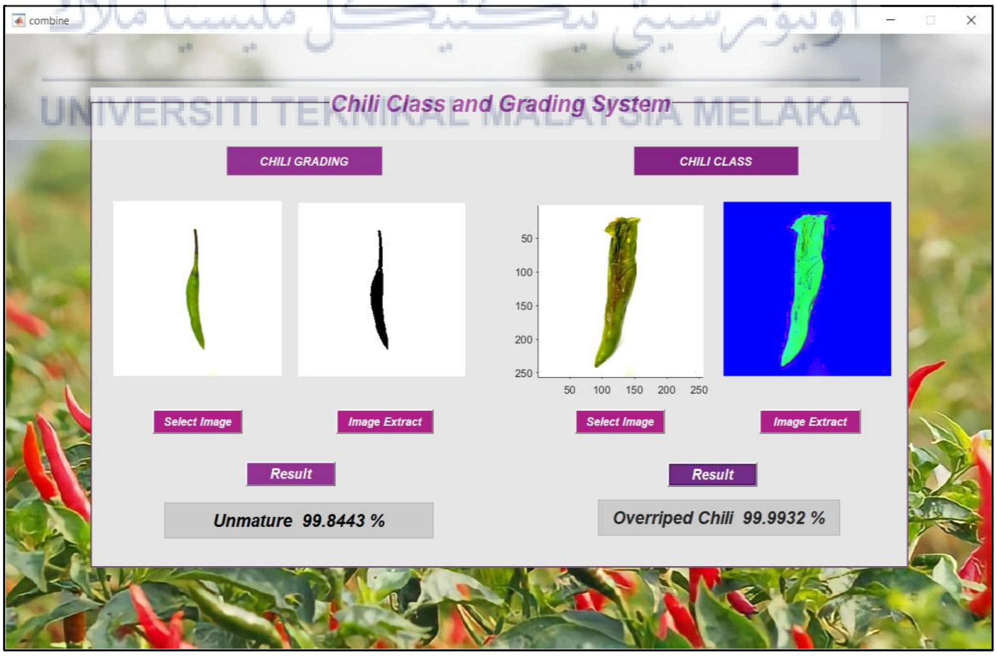
Green Chili Green Chili Overriped Chili Overriped Chili Red Chili

Figure 4.8: Classification for Variety Class of Chili Fruit

4.1.2.2 Graphical User Interfaces (GUI) System

A 3.11 GHz Intel i5-11300H quad-core processor, 8GB of RAM, and Windows 10 were used in this work. Figure 4.9 shows GUI using MATLAB Software for agricultural systems that can identify the maturity category and variety class of chili fruit.

The GUI consists of 3 components for maturity category and the variety of chili fruit. The first part is image selection for grading and class variety of chili image to the system. The second part is image extraction to detect the size and color of chili fruit. During this process, image is resized to 256×256 pixel. The training and testing images were then separated into 7:3 ratios. In the third step, results are obtained for the maturity category and variety class chili fruit. After pressing the result button, the accuracy for both maturity category and variety class image of chili fruit is shown in Figure 4.9.



Unmature 99.8443 %

Overriped Chili 99.9932 %

Figure 4.9: GUI System for Chili Class and Grading System

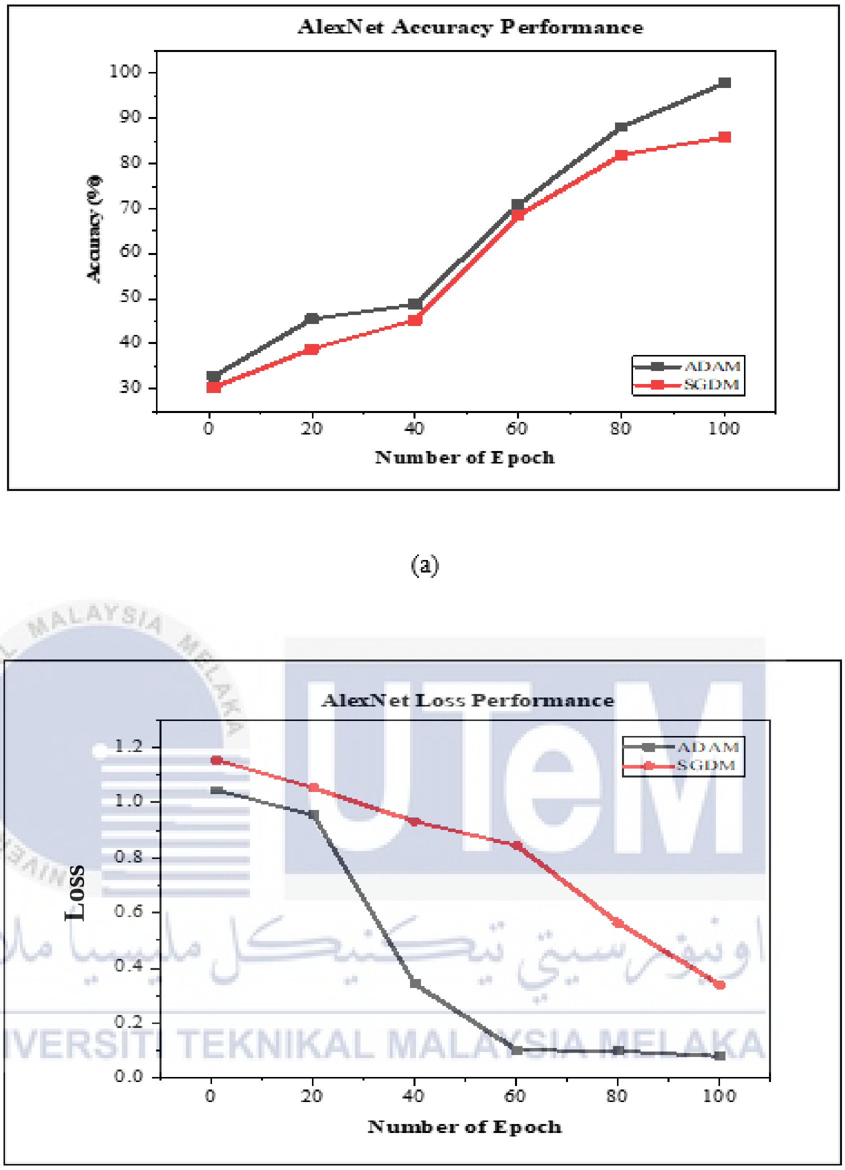
4.2 Analysis of Experiments

CNN has recently gained prominence due to its ability to recognize related properties for 2D images at numerous circumferences. CNN has proven in solving difficult problems promptly and accurately, while able to reduce the error rates [55]. Based on our observation, CNN is capable to estimate maturity and variety class chili fruit with high accuracy recognition using various architecture. Table 4.1 shows the parameter applied with various architectures including AlexNet, Inception ResNet version 2 and ResNet with 50 layers. Different optimizer such as SGDM and ADAM is used in this experiment with different value of learning rate.

4.2.1 Maturity Category of Chili Fruit

Figures 4.10 to 4.12 show the performance graphs of three architecture for maturity category of chili fruit in terms of accuracy and loss with learning rate of 0.0001. AlexNet, Inception ResNet version 2 and ResNet-50 achieved high accuracy with low learning rates using ADAM optimizer compared to the accuracy obtained by SGDM optimizer.

(a) AlexNet

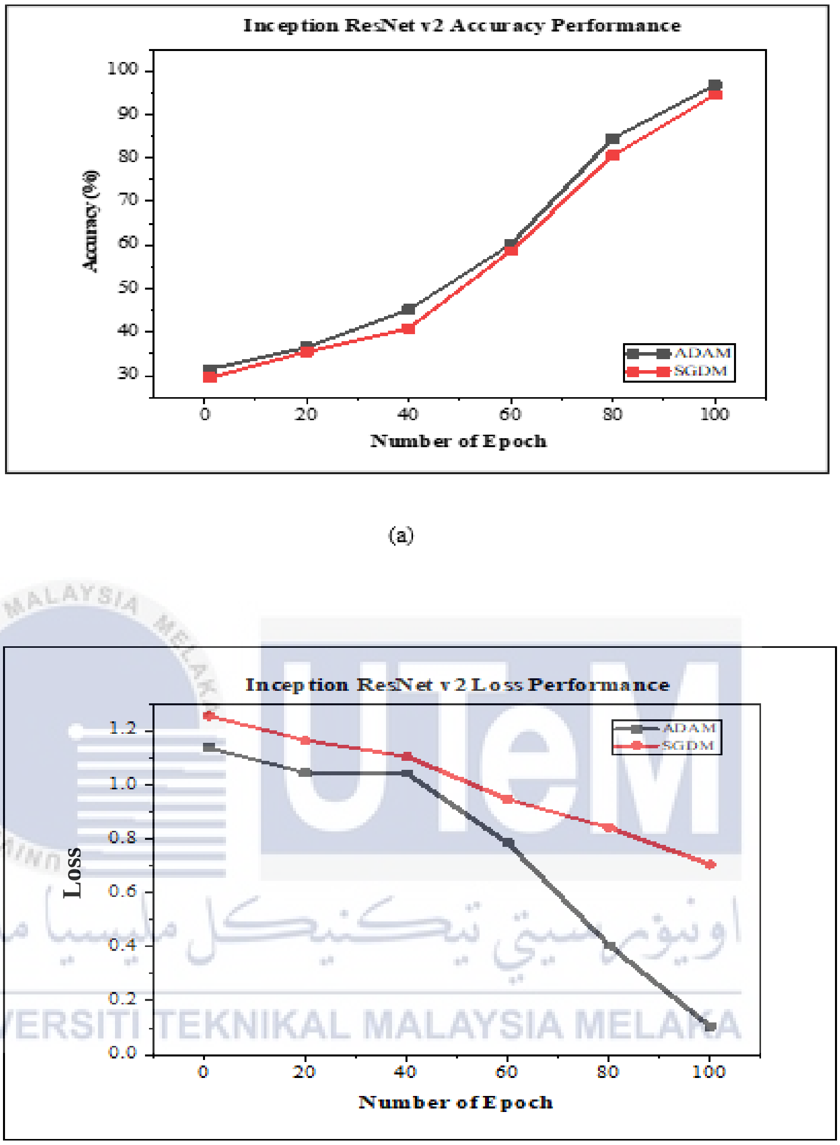


(a)

(b)

Figure 4.10: Performance Curves of AlexNet during Training with Learning Rate 0.0001 (a) Accuracy (b) Loss Function

(b) Inception ResNet Version 2

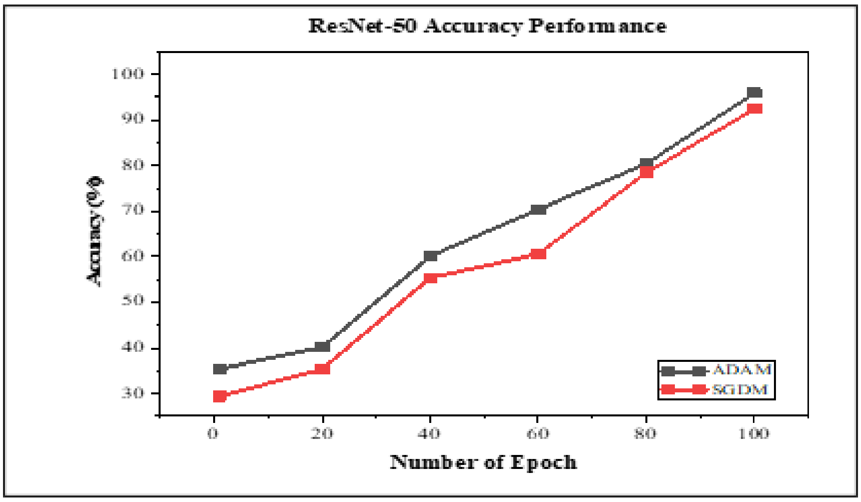


(a)

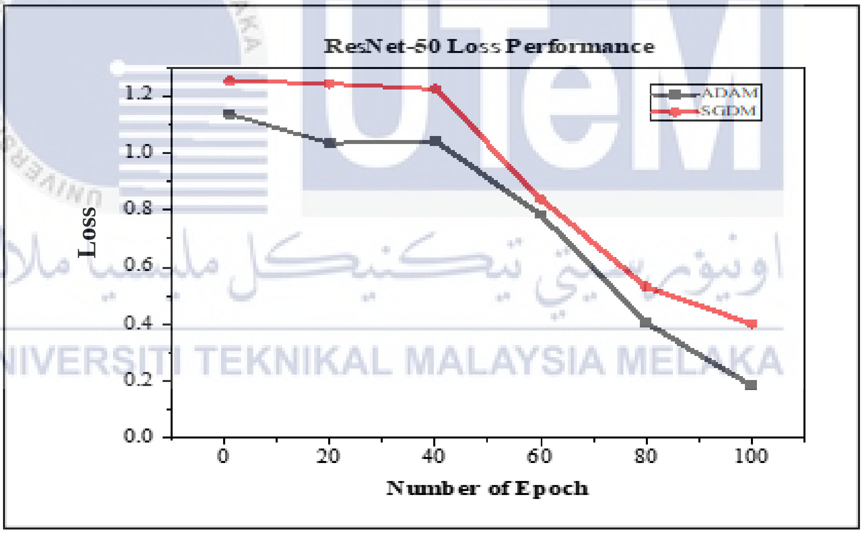
(b)

Figure 4.11: Performance Curves of Inception ResNet v2 during Training with Learning Rate 0.0001 (a) Accuracy (b) Loss Function

(c) ResNet-50



(a)



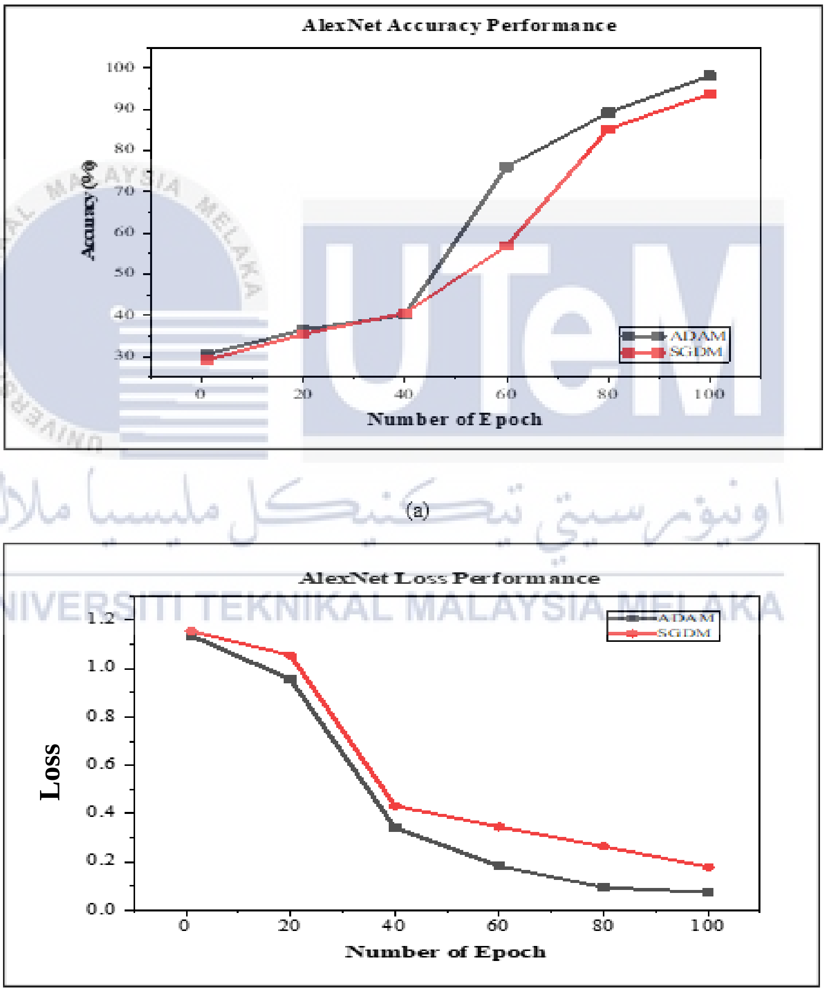
(b)

Figure 4.12: Performance Curves of ResNet-50 during Training with Learning Rate 0.0001 (a) Accuracy (b) Loss Function

4.2.2 Variety Class of Chili Fruit

The Figure 4.13 to 4.15 illustrates the performance of three architecture for variety class of chili fruit in terms of accuracy and the loss with learning rate of 0.0001. AlexNet, Inception ResNet version 2 and ResNet-50 produced utmost accuracy with low learning rates using ADAM optimizer compared to SGDM optimizer.

(a) AlexNet

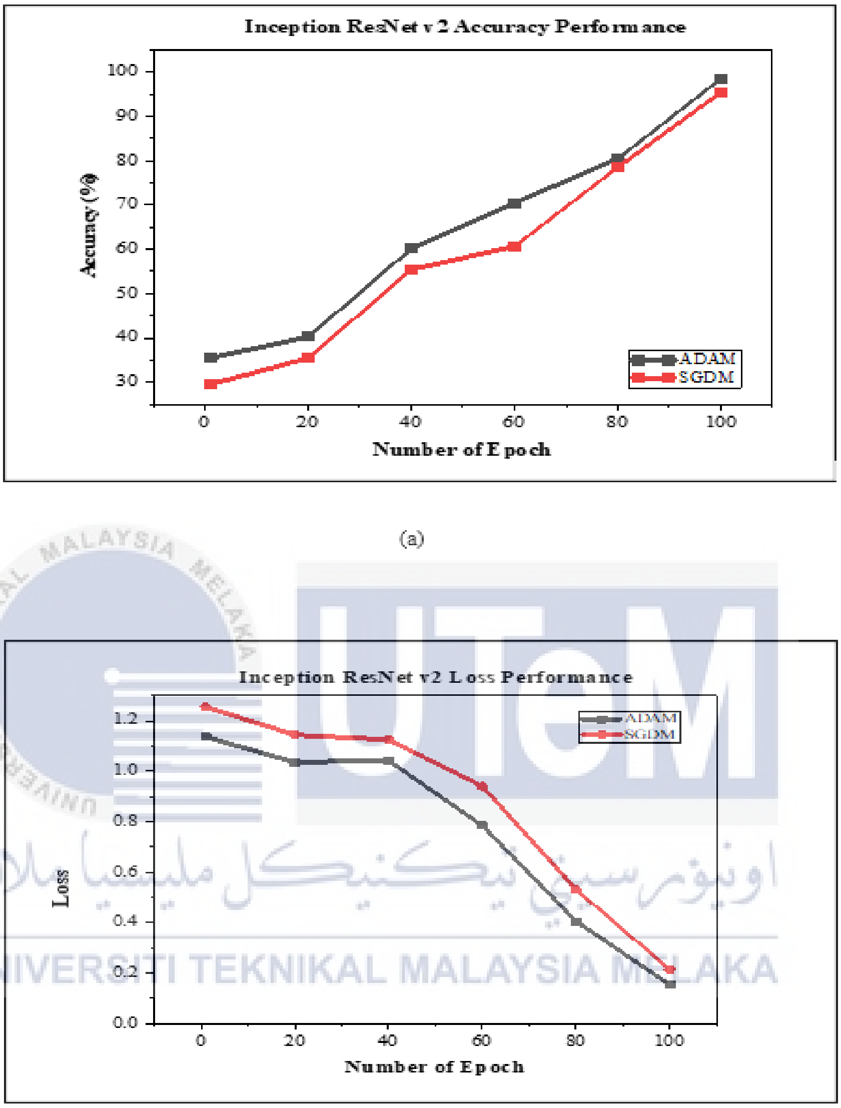


(a)

(b)

Figure 4.13: Performance Curves of AlexNet during Training with Learning Rate 0.0001 (a) Accuracy (b) Loss Function

(b) Inception ResNet Version 2

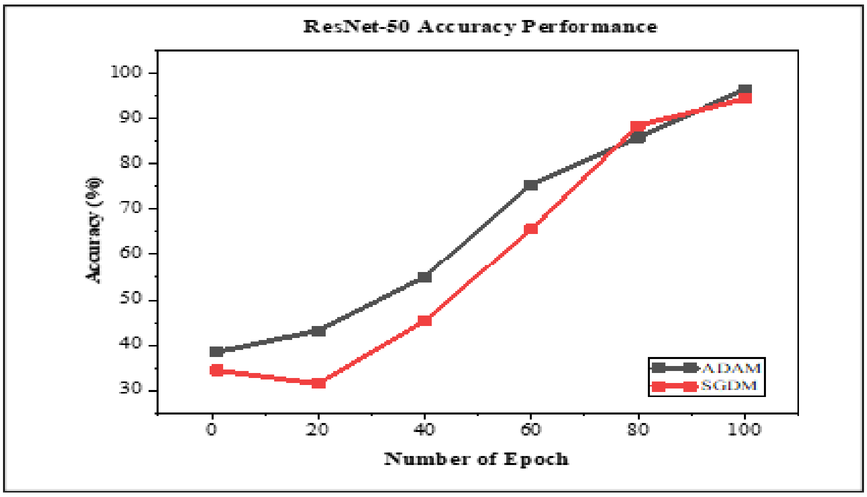


(a)

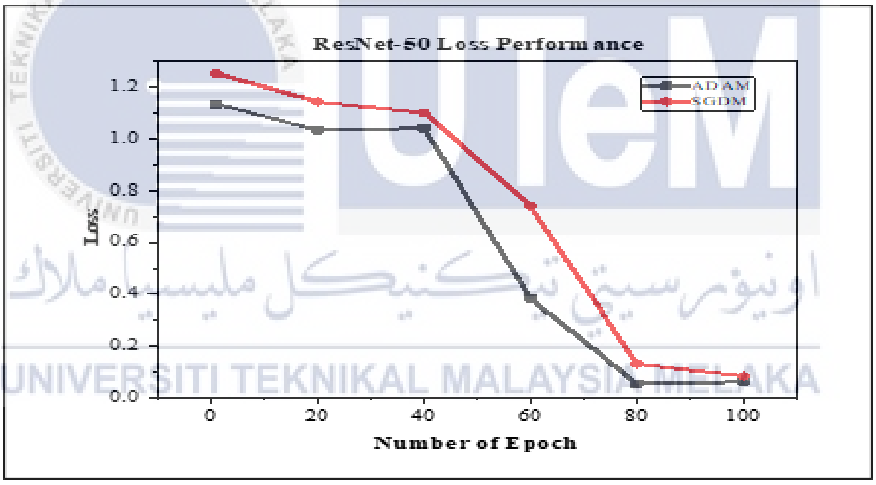
(b)

Figure 4.14: Performance Curves of Inception ResNet v2 during Training with Learning Rate 0.0001 (a) Accuracy (b) Loss Function

(c) ResNet-50



(a)



(b)

Figure 4.15: Performance Curves of ResNet-50 during Training with Learning Rate 0.0001 (a) Accuracy (b) Loss Function

4.3 Discussion of Overall Experiment

The performance of numerous architectures (AlexNet, Inception ResNet version 2 and ResNet-50) is evaluated in order to determine the maturity category and variety class. The experiments on two optimization algorithms; ADAM and SDGM with various architecture is explored in this work. The performance of the ADAM and SGDM could be improved by tuning the learning rate. As a result, ADAM and SGDM produced the greatest performance with learning rate of 0.0001.

ADAM optimizer and learning rate produced the best performance on the recognition of two categories of classes: maturity and variety of chili fruit.

Table 4.2 and 4.5 shows the accuracy obtained during the training process. AlexNet produced highest accuracy 97.88% with learning rate of 0.0001 using ADAM. However, the accuracy of SGDM with AlexNet achieved the accuracy of 85.89% with 0.0001 as a learning rate which is considered slightly lower than ADAM. The performance is also being measured in terms of run time where the lowest run time recorded is 2.42 minutes with 0.0001 as learning rate for ADAM and the highest run time of 6.40 recorded by SGDM with 0.01 learning rate using AlexNet. Most architectures showed superior performance when low learning rate is used. Overall, the most outstanding performance with all architectures recorded by ADAM is above 85% with less consumption time. Meanwhile, low accuracy is obtained by SGDM as shown in Figure 4.12.

Overall, it could be seen that ADAM optimizer is capable to produce high accuracy performance for both categories in comparison with SGDM optimizer. In terms of time taken for model evaluation, ADAM also reported has less time consumption than SGDM which is considered as much efficient for dealing with numerous of image classification problems. This result is clearly proven from the article reported by Rajakumari R and L. Kalaivani [56], a good performance can be achieved with different optimizers when the training is evaluated on different image datasets. The best architecture is recorded by AlexNet where the result has outperformed Inception ResNet v2 and ResNet-50 using ADAM optimizer in terms of accuracy. Less processing time is also to make this model is much effective and capable to detect the size and color of chili fruit with high accuracy.

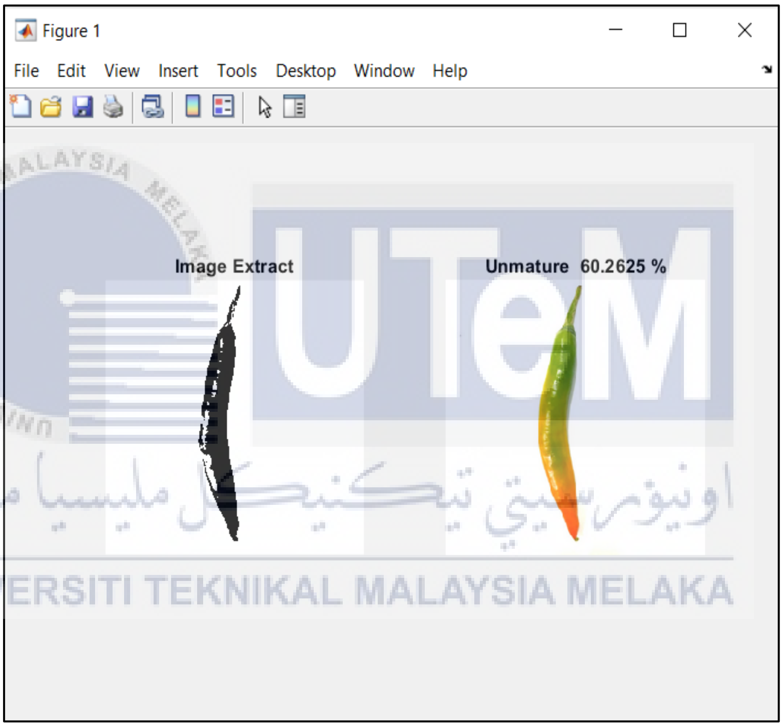


Figure 4.16: The Testing Performance using SGDM Optimizer

4.3.1 Maturity Category of Chili Fruit

(a) AlexNet

Table 4.2: AlexNet Performance for Maturity Category

Parameters	**Training Methods**	**Learning Rate**					
		0.01		0.001		0.0001	
		Training	Testing	Training	Testing	Training	Testing
Accuracy (%)	ADAM	90.62	89.57	95.88	94.12	**97.88**	**95.85**
Loss Rate		0.2129	0.2101	0.1421	0.1414	0.0796	0.0615
Runtime (minutes)		4.54		3.34		**2.42**	
Accuracy (%)	SGDM	79.55	75.55	81.89	80.88	85.89	83.33
Loss Rate		0.632	0.601	0.5521	0.4411	0.3395	0.2395
Runtime (minutes)		6.40		5.30		4.50	

(b) Inception ResNet Version 2

Table 4.3: Inception ResNet v2 Performance for Maturity Category

Parameters	**Training Methods**	**Learning Rate**					
		0.01		0.001		0.0001	
		Training	Testing	Training	Testing	Training	Testing
Accuracy (%)		74.29	72.09	94.29	92.34	**96.75**	**94.55**

Loss Rate	ADAM	0.4984	0.5234	0.3943	0.2023	0.1047	0.1156
Runtime (minutes)		3.32		2.59		**2.51**	
Accuracy (%)	SGDM	89.54	87.24	90.14	89.51	92.38	91.54
Loss Rate		1.5421	1.6452	1.4584	1.2452	0.7450	0.8452
Runtime (minutes)		5.45		5.12		4.55	

(c) ResNet-50

Table 4.4: ResNet-50 Performance for Maturity Category

Parameters	**Training Methods**	**Learning Rate**					
		0.01		0.001		0.0001	
		Training	Testing	Training	Testing	Training	Testing
Accuracy (%)	ADAM	88.41	87.45	90.45	89.54	**95.98**	**93.38**
Loss Rate		0.1756	0.2354	0.1654	0.2284	0.1856	0.2854
Runtime (minutes)		4.58		4.21		**3.32**	
Accuracy (%)	SGDM	85.44	83.42	89.54	89.44	92.45	89.64
Loss Rate		0.7546	0.6145	0.5421	0.4421	0.4021	0.4285
Runtime (minutes)		5.34		4.45		4.38	

4.3.2 Variety Class of Chili Fruit

(a) AlexNet

Table 4.5: AlexNet Performance for Variety Class

Parameters	**Training Methods**	**Learning Rate**					
		0.01		0.001		0.0001	
		Training	Testing	Training	Testing	Training	Testing
Accuracy (%)	ADAM	94.34	90.65	96.15	92.15	**98.11**	**94.44**
Loss Rate		0.0534	0.1340	0.0632	0.0981	0.0744	0.0855
Runtime (minutes)		4.53		4.32		**3.55**	
Accuracy (%)	SGDM	90.32	87.54	91.24	90.34	93.75	91.75
Loss Rate		0.2351	0.231	0.2125	0.224	0.1795	0.1835
Runtime (minutes)		5.20		4.55		4.30	

(b) Inception ResNet Version 2

Table 4.6: Inception ResNet v2 Performance for Variety Class

Parameters	**Training Methods**	**Learning Rate**					
		0.01		0.001		0.0001	
		Training	Testing	Training	Testing	Training	Testing
Accuracy (%)	ADAM	94.54	93.45	95.08	93.84	**97.54**	**96.54**
Loss Rate		0.2084	0.1954	0.1705	0.1845	0.1544	0.1745
Runtime (minutes)		3.45		3.11		**2.30**	

Accuracy (%)	SGDM	90.54	89.64	93.04	91.54	95.45	93.12
Loss Rate		0.8451	0.7541	0.7012	0.7181	0.2142	0.2134
Runtime (minutes)		5.28		4.58		4.28	

(c) ResNet-50

Table 4.7: ResNet-50 Performance for Variety Class

Parameters	**Training Methods**	**Learning Rate**					
		0.01		0.001		0.0001	
		Training	Testing	Training	Testing	Training	Testing
Accuracy (%)	ADAM	93.45	90.45	95.22	94.22	**96.54**	**95.22**
Loss Rate		0.0864	0.0842	0.0540	0.0565	0.0625	0.0752
Runtime (minutes)		4.10		3.54		**3.02**	
Accuracy (%)	SGDM	85.64	84.54	89.61	87.55	94.45	90.98
Loss Rate		0.1054	0.1154	0.09451	0.0930	0.0851	0.0842
Runtime (minutes)		5.29		4.56		4.22	

4.4 Environment and Sustainability

The agricultural industry has made enormous strides in recent years for conducting the process of detection and recognition in intelligent ways such as using autonomous robot and artificial intelligence. As an example, for identification of chili fruit using deep learning or CNN is invented from experts for improving the process of defining the variety and maturity of chili fruit.

An integration of several architectures such as AlexNet, InceptionResNetV2, and ResNet-50, high accuracy and less training time is obtained by ADAM optimizer. The architecture of InceptionResNetV2 demonstrated great success rates in identifying the maturity category and variety of chili fruit.

This novel approach is expected and believed is able to increase crop production by helping the farmers or industry partnership while lowering the costs by utilizing such available resources. This also contributes to agriculture grows aligned with the use of digital technology which enable farmers to make better solution while maximizing the harvesting.

In most practical ways, agriculture robots use gasoline which contributes to air pollution and might give poisonous to human and highly combustible to nature. The vapours are released when gasoline evaporates and those created materials will contributes to air pollution when gasoline is burned. So, this work not only gives a significant impact on agriculture industry, but also helps to save the environment by reducing air pollution.

CHAPTER 5

CONCLUSION AND FUTURE WORKS

This chapter describes the conclusion of the work that has been done and the summary of the achievements of the objective and analysis. This chapter also explains some suggestion for future work for further investigation.

5.1 Conclusion

In conclusion, the aim of this work has successfully achieved. As mentioned earlier, three objectives have been listed. The first objective is for measuring the length of the chili fruit by calculating from two points; calyx and apex in order to categorizing into three maturity categories; unmature, moderate mature and mature. A deep learning model is designed and implemented using by adapting the CNN to classify the maturity and variety class of chili fruit.

The second objective is to analyze the maturity category of chili fruit using ADAM and SGDM optimizer with the various architecture of CNN in terms of recognition accuracy. The proposed CNN with optimum parameter is designed and able to get the accuracy above 85% for various architecture. ADAM optimizer is able to record an excellent accuracy in comparison with SGDM optimizer in recognizing the chili size accurately.

The third objective is to classify and analyze the class varieties of chili fruit according to its color using various CNN architecture. The variety class is defined by detecting the color features and categorized it into three categories: green chili, red chili and overripe chili. The analysis is considered successful since this work is able to recognize and classify the chili according to its color with high accuracy.

The GUI system is built and intended to identify the grading and classification the variety class of chili fruit. The proposed GUI could become a medium for farmers for conducting their work such as to detect and to estimate the size and color of chili fruits in very short period of time.

In term of optimizer comparison, ADAM optimizer is considered as the best compared to SGDM in this case since the average performance using ADAM much higher than SGDM. This is due to ADAM optimizer combines the characteristics of Stochastic Gradient Descent with momentum and RMSProp techniques to deliver the best solution for sparse gradients. In addition, Inception ResNet version 2 is claimed as the best architecture since high accuracy is obtained in average for both experiments, maturity category and class varieties.

5.2 Future Work

2D images is used as dataset for to recognize and detection chili fruit since this category of image is considered as low production cost and easy to process. Image needs to be resized and it will make the process getting slower for detection. Somehow, there is some limitation of 2D image where the depth of the image is not presented which is considered as important factors for implementing the real-environment applications.

Hence, this work could be improved by using a depth camera to define the exactly position of chili from the plant. The depth camera is able produce the depth information which would present the distance from the camera to an object. Quick Measure is one of the preloaded applications to detect the object and displays its distance in real-time measurements. When there is an object is detected in the frame, 3D camera is operated by calculating its width, height, area, volume and other parameters as well. Therefore, by using the depth camera into this application, it will capture an image 3D viewpoint which is considered more accurate to estimate the size and recognize the color of the chili fruit according to its position.

REFERENCES

[1] K. Pawlak and M. Kołodziejczak, “The role of agriculture in ensuring food security in developing countries: Considerations in the context of the problem of sustainable food production,” *Sustain.*, vol. 12, no. 13, 2020, doi: 10.3390/su12135488.

[2] M. L. Praburaj, “Role of Agriculture in the Economic Development of a Country,” *Int. J. Commer.*, vol. 6, no. 3, p. 2, 2018, doi: 10.5281/zenodo.1323056.

[3] “Spatial features.” http://www.biomedware.com/files/documentation/boundaryseer/Projects/Spatial_feature_files.htm.

[4] A. Kavitha, “Deep Learning for Smart Agriculture,” vol. 9, no. 5, pp. 132–134, 2021.

[5] A. Habeeb, “Artificial intelligence Ahmed Habeeb University of Mansoura,” *Res. Gate*, vol. 7, no. 2, 2017, doi: 10.13140/RG.2.2.25350.88645/1.

[6] C. Law, “(PDF) Artificial Intelligence Definition, Ethics and Standards,” 2019,

[Online]. Available: https://www.researchgate.net/publication/332548325_Artificial_Intelligence_Definition_Ethics_and_Standards.

[7] R. Ashri, "What Is AI?," *AI-Powered Work.*, pp. 15–29, 2020, doi: 10.1007/978-1-4842-5476-9_2.

[8] B. W. Wirtz, J. C. Weyerer, and C. Geyer, "Artificial Intelligence and the Public Sector—Applications and Challenges," *Int. J. Public Adm.*, vol. 42, no. 7, pp. 596–615, 2019, doi: 10.1080/01900692.2018.1498103.

[9] D. Alfer'ev, "Artificial intelligence in agriculture," *Agric. Lifestock Technol. / АгроЗооТехника*, vol. 7, no. 4 (4), 2018, doi: 10.15838/alt.2018.1.4.5.

[10] P. Dönmez, "Introduction to Machine Learning, 2nd ed., by Ethem Alpaydın. Cambridge, MA: The MIT Press2010. ISBN: 978-0-262-01243-0. $54/£ 39.95 + 584 pages.," *Nat. Lang. Eng.*, vol. 19, no. 2, pp. 285–288, 2013, doi: 10.1017/s1351324912000290.

[11] M. Mohammed, M. B. Khan, and E. B. M. Bashie, *Machine learning: Algorithms and applications*, no. July. 2016.

[12] IBM, "Supervised Learning," *Cloud Education*. https://www.ibm.com/cloud/learn/supervised-learning (accessed May 29, 2021).

[13] J. Fidler Dennis and L. Arnroth, "Supervised Learning Techniques:A comparison of the Random Forest and the Support Vector Machine," 2015.

[14] Q. Liu and Y. Wu, “Encyclopedia of the Sciences of Learning,” *Encycl. Sci. Learn.*, no. January 2012, 2012, doi: 10.1007/978-1-4419-1428-6.

[15] P. Dayan, “Unsupervised Learning,” *Encycl. Neurosci.*, pp. 4154–4154, 2008, doi: 10.1007/978-3-540-29678-2_6202.

[16] Divyansh Dwivedi, “Machine Learning For Beginners,” *data science*. https://towardsdatascience.com/machine-learning-for-beginners-d247a9420dab (accessed May 30, 2021).

[17] IBM Education, “What is Unsupervised Learning? | IBM,” *Ibm*, no. August 2018, pp. 1–8, 2020, doi: 10.13140/RG.2.2.33325.10720.

[18] “Reinforcement Learning: What is, Algorithms, Applications, Example.” https://www.guru99.com/reinforcement-learning-tutorial.html.

[19] S. R. Hinton GE, “Reducing the dimensionality of data with neural networks,” *Science*, 2006. .

[20] H. Mureşan and M. Oltean, “Fruit recognition from images using deep learning,” *arXiv*, no. June 2018, 2017, doi: 10.2478/ausi-2018-0002.

[21] H. Tachibana, “葉月ちゃんでもできる Deep Learning,” *Sig2D’14*, p. 1, 2014, [Online]. Available: http://files.sig2d.org/sig2d14.pdf#page=5.

[22] Artem Oppermann, “What is Deep Learning and How does it work?,” *Towards Data Science*, 2019. https://towardsdatascience.com/what-is-deep-learning-and-how-does-it-work-2ce44bb692ac (accessed May 24, 2021).

[23] J. Ahmad, H. Farman, and Z. Jan, *Deep Learning Methods and Applications BT*

- *Deep Learning: Convergence to Big Data Analytics*. Springer Singapore, 2019.

[24] JiTU7, “Introduction to Supervised Deep Learning Algorithms!,” *Analytics Vidhya*, 2021. https://www.analyticsvidhya.com/blog/2021/05/introduction-to-supervised-deep-learning-algorithms/ (accessed Jun. 16, 2021).

[25] 山下隆義, “Convolutional Neural Network による画像認識と視覚的説明,” 人工知能学会全国大会論文集 一般社団法人 人工知能 …, pp. 1–68, 2019, [Online]. Available: https://www.jstage.jst.go.jp/article/pjsai/JSAI2019/0/JSAI2019_3E3OS12a01/_article/-char/ja/.

[26] V. Meel, “ANN and CNN: Analyzing Differences and Similarities,” *viso.ai*. https://viso.ai/deep-learning/ann-and-cnn-analyzing-differences-and-similarities/.

[27] IBM Education, “Recurrent Neural Networks,” *IBM*, 2020. https://www.ibm.com/cloud/learn/recurrent-neural-networks.

[28] A. K. Jain, “Data clustering: 50 years beyond K-means,” *Pattern Recognit. Lett.*, vol. 31, no. 8, pp. 651–666, 2010, doi: 10.1016/j.patrec.2009.09.011.

[29] S. Kotsiantis and D. Kanellopoulos, “Association Rules Mining: A Recent Overview,” *Science (80-.).*, vol. 32, no. 1, pp. 71–82, 2006.

[30] L. Morissette and S. Chartier, “The k-means clustering technique: General considerations and implementation in Mathematica,” *Tutor. Quant. Methods*

Psychol., vol. 9, no. 1, pp. 15–24, 2013, doi: 10.20982/tqmp.09.1.p015.

[31] A. Ghosh, A. Sufian, F. Sultana, A. Chakrabarti, and D. De, *Fundamental concepts of convolutional neural network*, vol. 172, no. January. 2019.

[32] Y. D. Zhang *et al.*, “Image based fruit category classification by 13-layer deep convolutional neural network and data augmentation,” *Multimed. Tools Appl.*, vol. 78, no. 3, pp. 3613–3632, 2019, doi: 10.1007/s11042-017-5243-3.

[33] A. Mathew, P. Amudha, and S. Sivakumari, “Deep learning techniques: an overview,” *Adv. Intell. Syst. Comput.*, vol. 1141, no. January, pp. 599–608, 2021, doi: 10.1007/978-981-15-3383-9_54.

[34] Z. M. Khaing, Y. Naung, and P. H. Htut, “Development of control system for fruit classification based on convolutional neural network,” *Proc. 2018 IEEE Conf. Russ. Young Res. Electr. Electron. Eng. ElConRus 2018*, vol. 2018-Janua, pp. 1805–1807, 2018, doi: 10.1109/EIConRus.2018.8317456.

[35] MATLAB, “Deep Learning Toolbox,” *MathWorks*. https://www.mathworks.com/products/deep-learning.html#:~:text=Deep Learning Toolbox™ provides,%2C pretrained models%2C and apps.&text=The Experiment Manager app helps,compare code from different experiments. (accessed May 30, 2021).

[36] P. Kim, *MATLAB Deep Learning: With Machine Learning, Neural Networks and Artificial Intelligence*. 2017.

[37] S. R. Dubey and A. S. Jalal, “Fruit and vegetable recognition by fusing colour and texture features of the image using machine learning,” *Int. J. Appl. Pattern*

Recognit., vol. 2, no. 2, p. 160, 2015, doi: 10.1504/ijapr.2015.069538.

[38] T. Ishikawa *et al.*, “Classification of strawberry fruit shape by machine learning,” *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci. - ISPRS Arch.*, vol. 42, no. 2, pp. 463–470, 2018, doi: 10.5194/isprs-archives-XLII-2-463-2018.

[39] M. Ataş, Y. Yardimci, and A. Temizel, “A new approach to aflatoxin detection in chili pepper by machine vision,” *Comput. Electron. Agric.*, vol. 87, pp. 129–141, 2012, doi: 10.1016/j.compag.2012.06.001.

[40] O. Cruz-Domínguez *et al.*, “A novel method for dried chili pepper classification using artificial intelligence,” *J. Agric. Food Res.*, vol. 3, no. October 2020, p. 100099, 2021, doi: 10.1016/j.jafr.2021.100099.

[41] M. Kaur and R. Sharma, “Quality Detection of Fruits by Using ANN Technique,” *IOSR J. Electron. Commun. Eng. Ver. II*, vol. 10, no. 4, pp. 2278–2834, 2015, doi: 10.9790/2834-10423541.

[42] A. Taofik, N. Ismail, Y. A. Gerhana, K. Komarujaman, and M. A. Ramdhani, “Design of Smart System to Detect Ripeness of Tomato and Chili with New Approach in Data Acquisition,” *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 288, no. 1, pp. 0–6, 2018, doi: 10.1088/1757-899X/288/1/012018.

[43] S. Deepika, “FRUIT MATURITY AND DISEASE DETECTION USING ARTIFICIAL NEURAL NETWORK,” no. 09, pp. 144–151, 2020.

[44] N. Fadilah, J. M. Saleh, H. Ibrahim, and Z. A. Halim, “Oil palm fresh fruit bunch ripeness classification using artificial neural network,” *ICIAS 2012 -*

2012 4th Int. Conf. Intell. Adv. Syst. A Conf. World Eng. Sci. Technol. Congr. - Conf. Proc., vol. 1, pp. 18–21, 2012, doi: 10.1109/ICIAS.2012.6306151.

[45] N. S. Naganhalli and S. Terdal, “Network intrusion detection using supervised machine learning technique,” *Int. J. Sci. Technol. Res.*, vol. 8, no. 9, pp. 345–350, 2019.

[46] T. Bezdan and N. Bačanin Džakula, “Convolutional Neural Network Layers and Architectures,” no. January, pp. 445–451, 2019, doi: 10.15308/sinteza-2019-445-451.

[47] H. Alaskar, A. Hussain, N. Al-Aseem, P. Liatsis, and D. Al-Jumeily, “Application of convolutional neural networks for automated ulcer detection in wireless capsule endoscopy images,” *Sensors (Switzerland)*, vol. 19, no. 6, 2019, doi: 10.3390/s19061265.

[48] L. D. Nguyen, D. Lin, Z. Lin, and J. Cao, “Deep CNNs for microscopic image classification by exploiting transfer learning and feature concatenation,” *Proc. - IEEE Int. Symp. Circuits Syst.*, vol. 2018-May, no. June, 2018, doi: 10.1109/ISCAS.2018.8351550.

[49] V. Sangeetha and K. J. R. Prasad, “Syntheses of novel derivatives of 2-acetylfuro[2,3-a]carbazoles, benzo[1,2-b]-1,4-thiazepino[2,3-a]carbazoles and 1-acetyloxycarbazole-2- carbaldehydes,” *Indian J. Chem. - Sect. B Org. Med. Chem.*, vol. 45, no. 8, pp. 1951–1954, 2006, doi: 10.1002/chin.200650130.

[50] Matlab, “Image Filtering,” *MathWorks*. https://www.mathworks.com/help/images/linear-filtering.html.

[51] P. Nandhini and J. Jaya, “Image Segmentation for Food Quality Evaluation Using Computer Vision System,” vol. 4, no. 2, pp. 1–3, 2014.

[52] “Convolutional Neural Networks.” https://www.ibm.com/cloud/learn/convolutional-neural-networks.

[53] S. Albawi, T. A. Mohammed, and S. Al-Zawi, “Understanding of a convolutional neural network,” *Proc. 2017 Int. Conf. Eng. Technol. ICET 2017*, vol. 2018-Janua, no. April 2018, pp. 1–6, 2018, doi: 10.1109/ICEngTechnol.2017.8308186.

[54] C. Pelletier, G. Webb, and F. Petitjean, “Temporal Convolutional Neural Network for the Classification of Satellite Image Time Series,” *Remote Sens.*, vol. 11, p. 523, Mar. 2019, doi: 10.3390/rs11050523.

[55] S. M. Hassan, A. K. Maji, M. Jasiński, Z. Leonowicz, and E. Jasińska, “Identification of plant-leaf diseas[1] S. M. Hassan, A. K. Maji, M. Jasiński, Z. Leonowicz, and E. Jasińska, ‘Identification of plant-leaf diseases using cnn and transfer-learning approach,’ Electron., vol. 10, no. 12, 2021, doi: 10.3390/electronics101213,” *Electron.*, vol. 10, no. 12, 2021.

[56] R. Rajakumari, “Breast Cancer Detection and Classification using Deeper Convolutional Neural Networks based on Wavelet Packet Decomposition Techniques,” 2021.

[57] Hunter Heidenreich, “What are the types of machine learning?,” *data science*, 2018. https://towardsdatascience.com/what-are-the-types-of-machine-learning-e2b9e5d1756f (accessed Jun. 15, 2021).

[58] "Unsupervised Learning – Machine Learning Algorithms," *TechVidvan*. https://techvidvan.com/tutorials/unsupervised-learning/ (accessed Jun. 15, 2021).

[59] Matlab, "What Is Reinforcement Learning?," *MathWorks*. https://www.mathworks.com/discovery/reinforcement-learning.html (accessed Jun. 16, 2021).

[60] The Mathworks, "Introducing Deep Learning with MATLAB: 'What is Deep Learning?,'" 2018, [Online]. Available: https://ch.mathworks.com/campaigns/offers/deep-learning-with-matlab.html?elqCampaignId=10588%0Ahttps://www.mathworks.com/content/dam/mathworks/ebook/gated/80879v00_Deep_Learning_ebook.pdf.

[61] S. Albelwi and A. Mahmood, "A framework for designing the architectures of deep Convolutional Neural Networks," *Entropy*, vol. 19, no. 6, 2017, doi: 10.3390/e19060242.

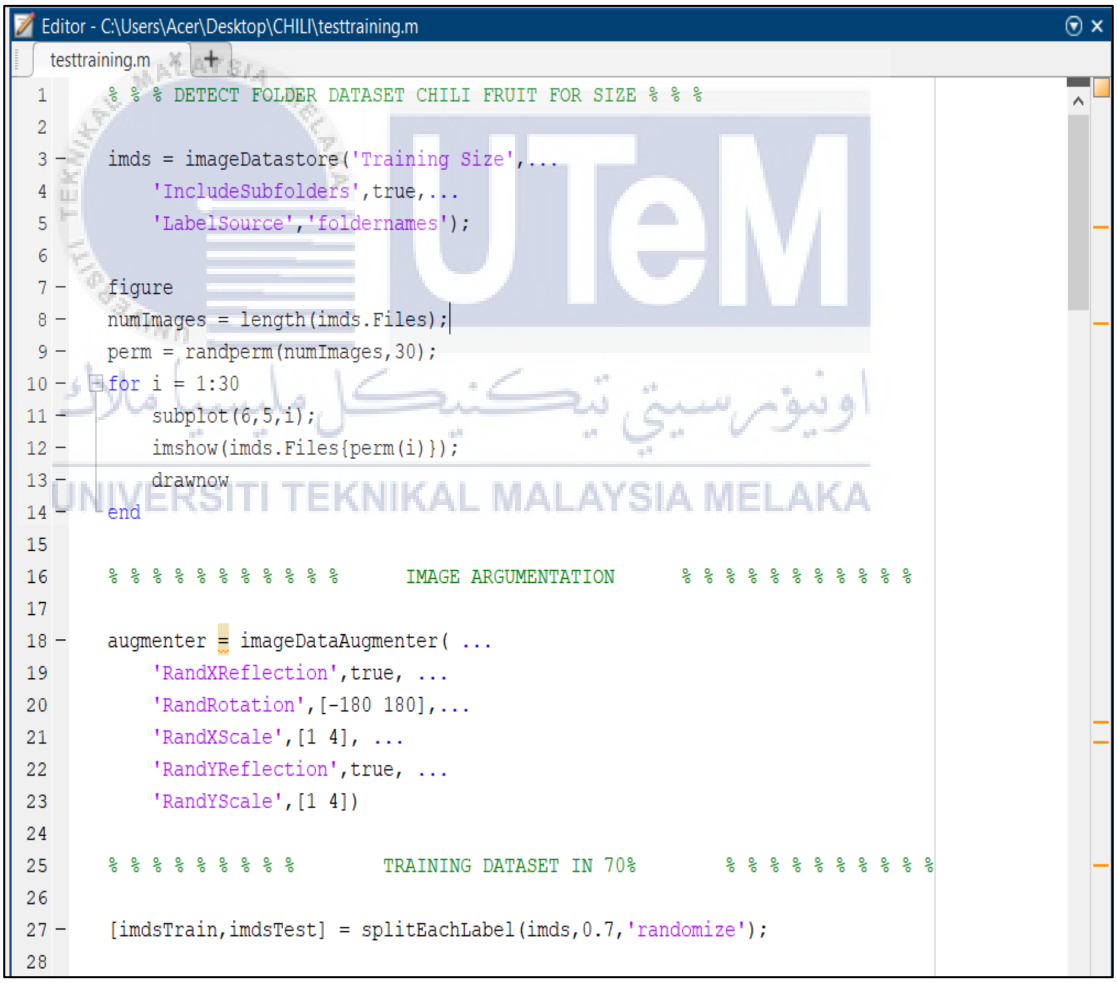
[62] J. Llamas, P. M. Lerones, R. Medina, E. Zalama, and J. Gómez-García-Bermejo, "Classification of architectural heritage images using deep learning techniques," *Appl. Sci.*, vol. 7, no. 10, pp. 1–26, 2017, doi: 10.3390/app7100992.

[63] Matt Bray, "Pepper Anatomy: What's Inside Your Chili?" https://www.pepperscale.com/pepper-anatomy/.

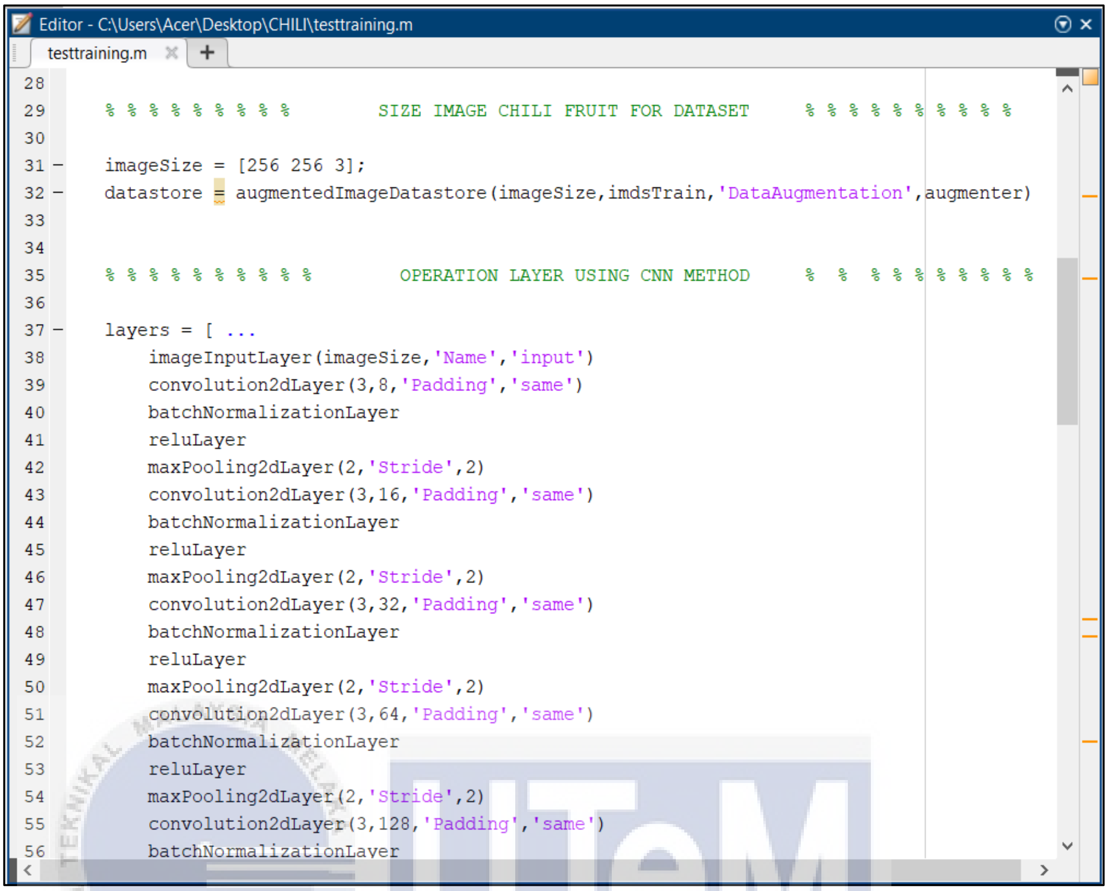
APPENDICES

APPENDIX A

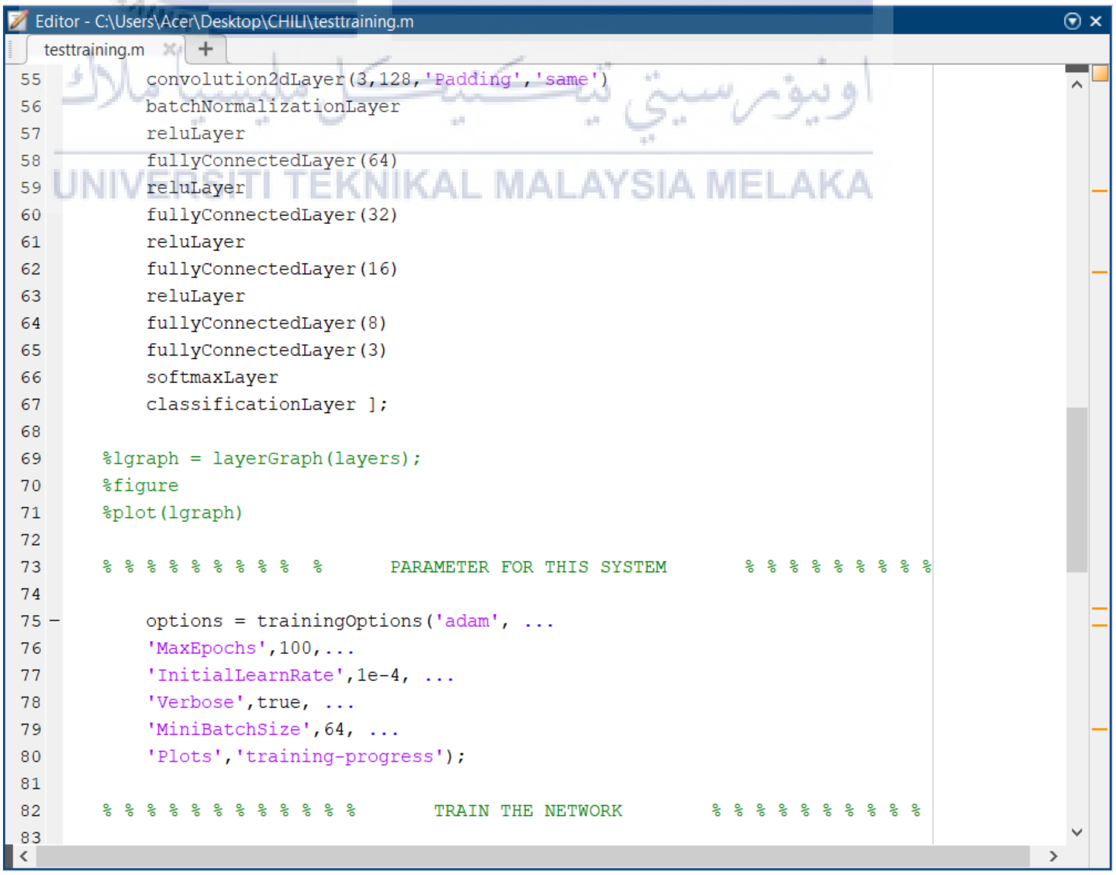
CODING FOR MATURITY CATEGORY CHILI FRUIT



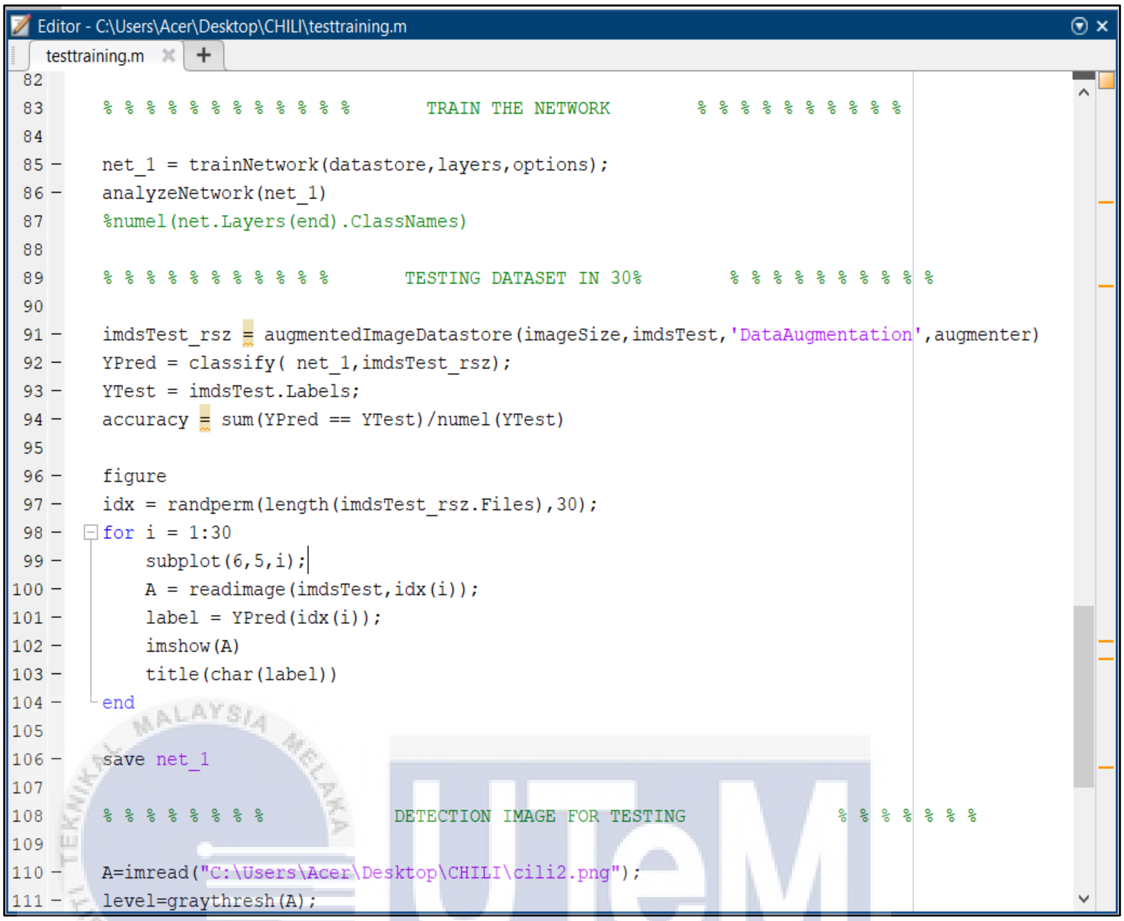
```
% % % DETECT FOLDER DATASET CHILI FRUIT FOR SIZE % % %

imds = imageDatastore('Training Size',...
    'IncludeSubfolders',true,...
    'LabelSource','foldernames');

figure
numImages = length(imds.Files);
perm = randperm(numImages,30);
for i = 1:30
    subplot(6,5,i);
    imshow(imds.Files{perm(i)});
    drawnow
end

% % % % % % % % % % %      IMAGE ARGUMENTATION      % % % % % % % % % % %

augmenter = imageDataAugmenter( ...
    'RandXReflection',true, ...
    'RandRotation',[-180 180],...
    'RandXScale',[1 4], ...
    'RandYReflection',true, ...
    'RandYScale',[1 4])

% % % % % % % % %        TRAINING DATASET IN 70%       % % % % % % % % % %

[imdsTrain,imdsTest] = splitEachLabel(imds,0.7,'randomize');

```



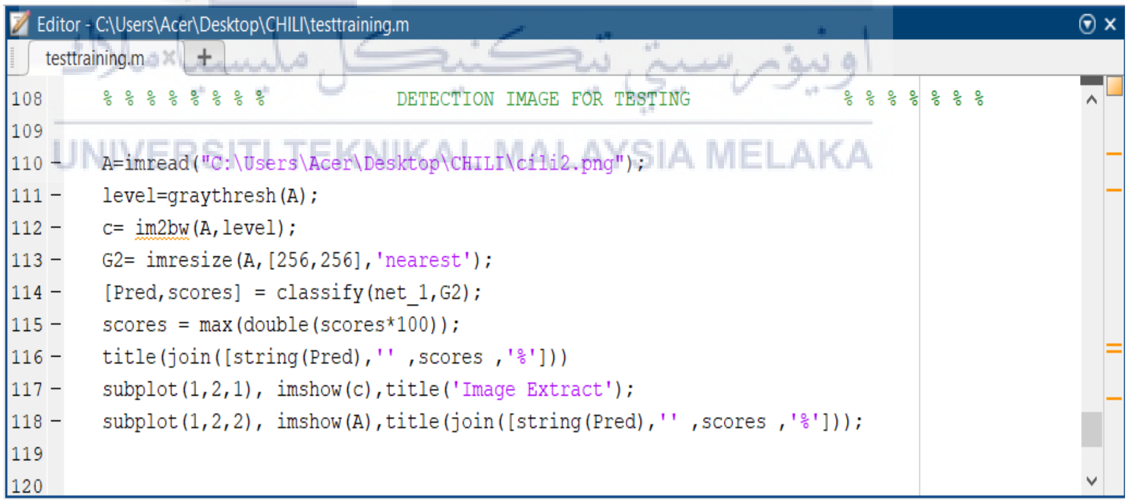
```

% % % % % % % % %        SIZE IMAGE CHILI FRUIT FOR DATASET     % % % % % % % % % %

imageSize = [256 256 3];
datastore = augmentedImageDatastore(imageSize,imdsTrain,'DataAugmentation',augmenter)


% % % % % % % % % %        OPERATION LAYER USING CNN METHOD     %  %  % % % % % % % %

layers = [ ...
    imageInputLayer(imageSize,'Name','input')
    convolution2dLayer(3,8,'Padding','same')
    batchNormalizationLayer
    reluLayer
    maxPooling2dLayer(2,'Stride',2)
    convolution2dLayer(3,16,'Padding','same')
    batchNormalizationLayer
    reluLayer
    maxPooling2dLayer(2,'Stride',2)
    convolution2dLayer(3,32,'Padding','same')
    batchNormalizationLayer
    reluLayer
    maxPooling2dLayer(2,'Stride',2)
    convolution2dLayer(3,64,'Padding','same')
    batchNormalizationLayer
    reluLayer
    maxPooling2dLayer(2,'Stride',2)
    convolution2dLayer(3,128,'Padding','same')
    batchNormalizationLayer
```



```
    convolution2dLayer(3,128,'Padding','same')
    batchNormalizationLayer
    reluLayer
    fullyConnectedLayer(64)
    reluLayer
    fullyConnectedLayer(32)
    reluLayer
    fullyConnectedLayer(16)
    reluLayer
    fullyConnectedLayer(8)
    fullyConnectedLayer(3)
    softmaxLayer
    classificationLayer ];

%lgraph = layerGraph(layers);
%figure
%plot(lgraph)

% % % % % % % % %  %       PARAMETER FOR THIS SYSTEM       % % % % % % % % %

    options = trainingOptions('adam', ...
    'MaxEpochs',100,...
    'InitialLearnRate',1e-4, ...
    'Verbose',true, ...
    'MiniBatchSize',64, ...
    'Plots','training-progress');

% % % % % % % % % % % %        TRAIN THE NETWORK        % % % % % % % % % %

```



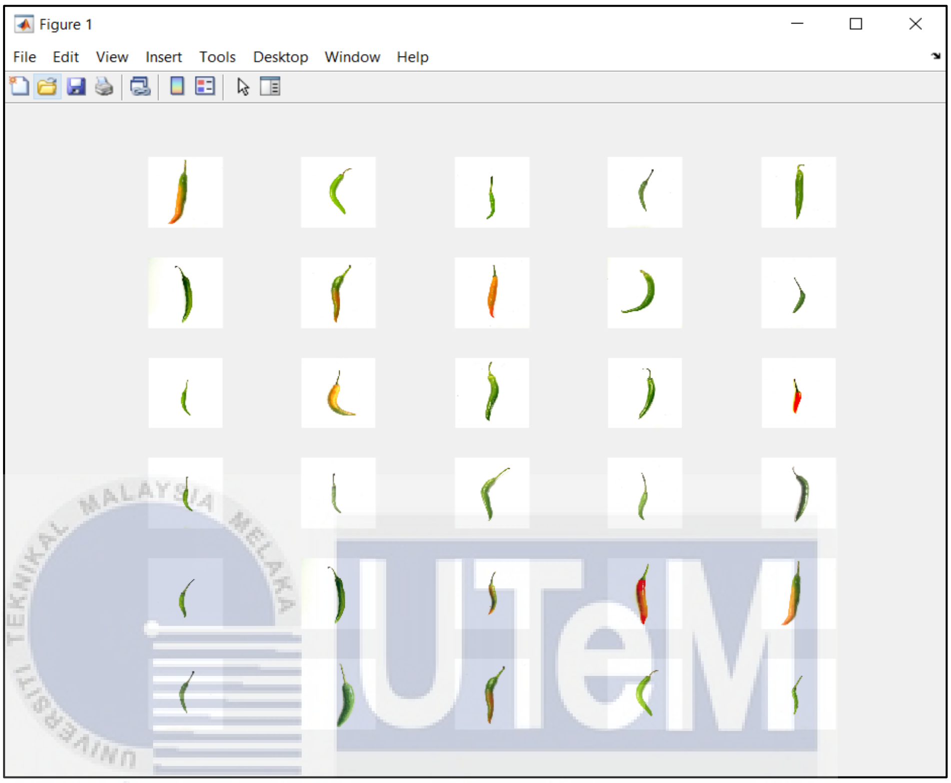
```
% % % % % % % % % % % %      TRAIN THE NETWORK      % % % % % % % % % %

net_1 = trainNetwork(datastore,layers,options);
analyzeNetwork(net_1)
%numel(net.Layers(end).ClassNames)

% % % % % % % % % % %      TESTING DATASET IN 30%      % % % % % % % % % %

imdsTest_rsz = augmentedImageDatastore(imageSize,imdsTest,'DataAugmentation',augmenter)
YPred = classify( net_1,imdsTest_rsz);
YTest = imdsTest.Labels;
accuracy = sum(YPred == YTest)/numel(YTest)

figure
idx = randperm(length(imdsTest_rsz.Files),30);
for i = 1:30
    subplot(6,5,i);
    A = readimage(imdsTest,idx(i));
    label = YPred(idx(i));
    imshow(A)
    title(char(label))
end

save net_1

% % % % % % % %          DETECTION IMAGE FOR TESTING          % % % % % % %

A=imread("C:\Users\Acer\Desktop\CHILI\cili2.png");
level=graythresh(A);
```

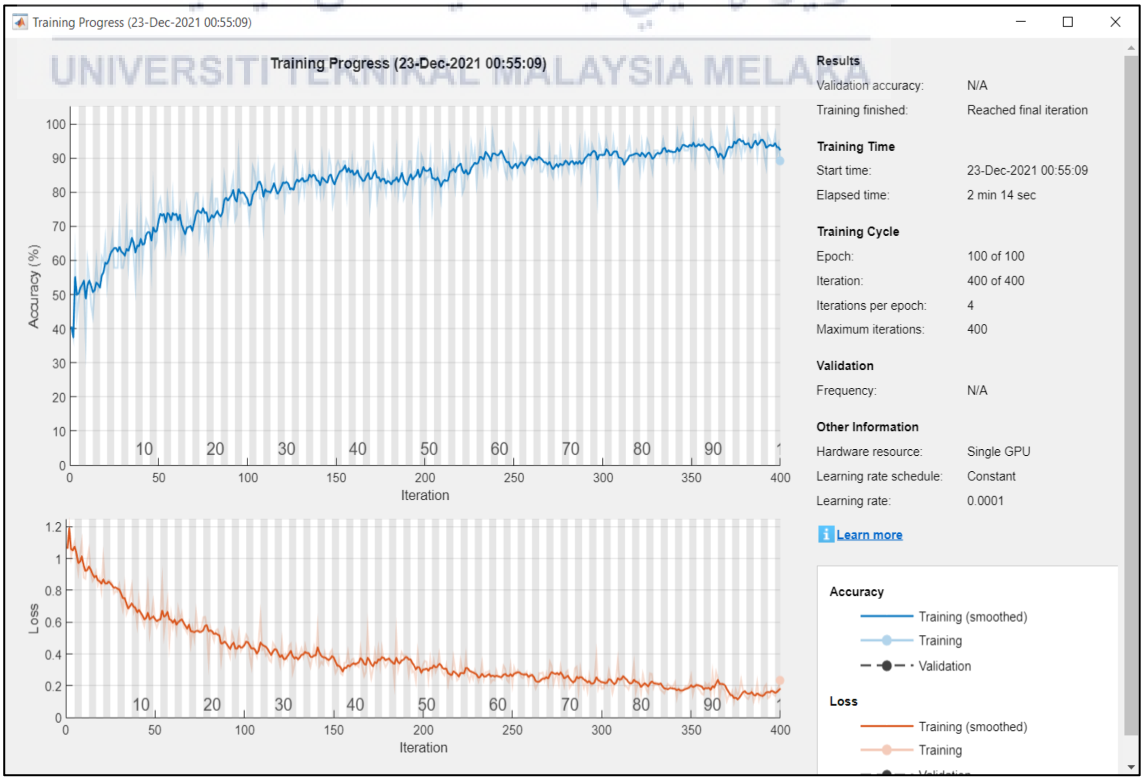


```
% % % % % % % %          DETECTION IMAGE FOR TESTING          % % % % % % %

A=imread("C:\Users\Acer\Desktop\CHILI\cili2.png");
level=graythresh(A);
c= im2bw(A,level);
G2= imresize(A,[256,256],'nearest');
[Pred,scores] = classify(net_1,G2);
scores = max(double(scores*100));
title(join([string(Pred),'' ,scores ,'%']))
subplot(1,2,1), imshow(c),title('Image Extract');
subplot(1,2,2), imshow(A),title(join([string(Pred),'' ,scores ,'%']));
```

APPENDIX B

RUN CODING FOR MATURITY CATEGORY





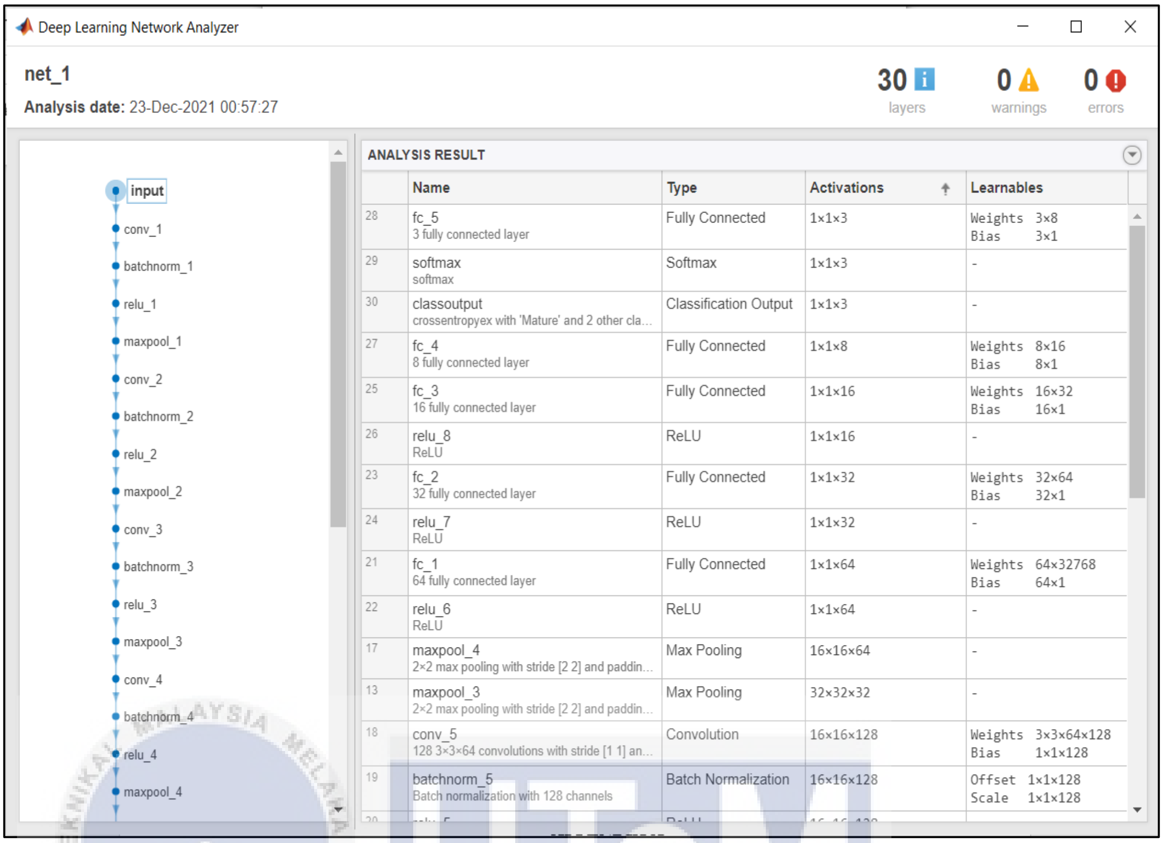
Results
Validation accuracy: N/A
Training finished: Reached final iteration

Training Time
Start time: 23-Dec-2021 00:55:09
Elapsed time: 2 min 14 sec

Training Cycle
Epoch: 100 of 100
Iteration: 400 of 400
Iterations per epoch: 4
Maximum iterations: 400

Validation
Frequency: N/A

Other Information
Hardware resource: Single GPU
Learning rate schedule: Constant
Learning rate: 0.0001



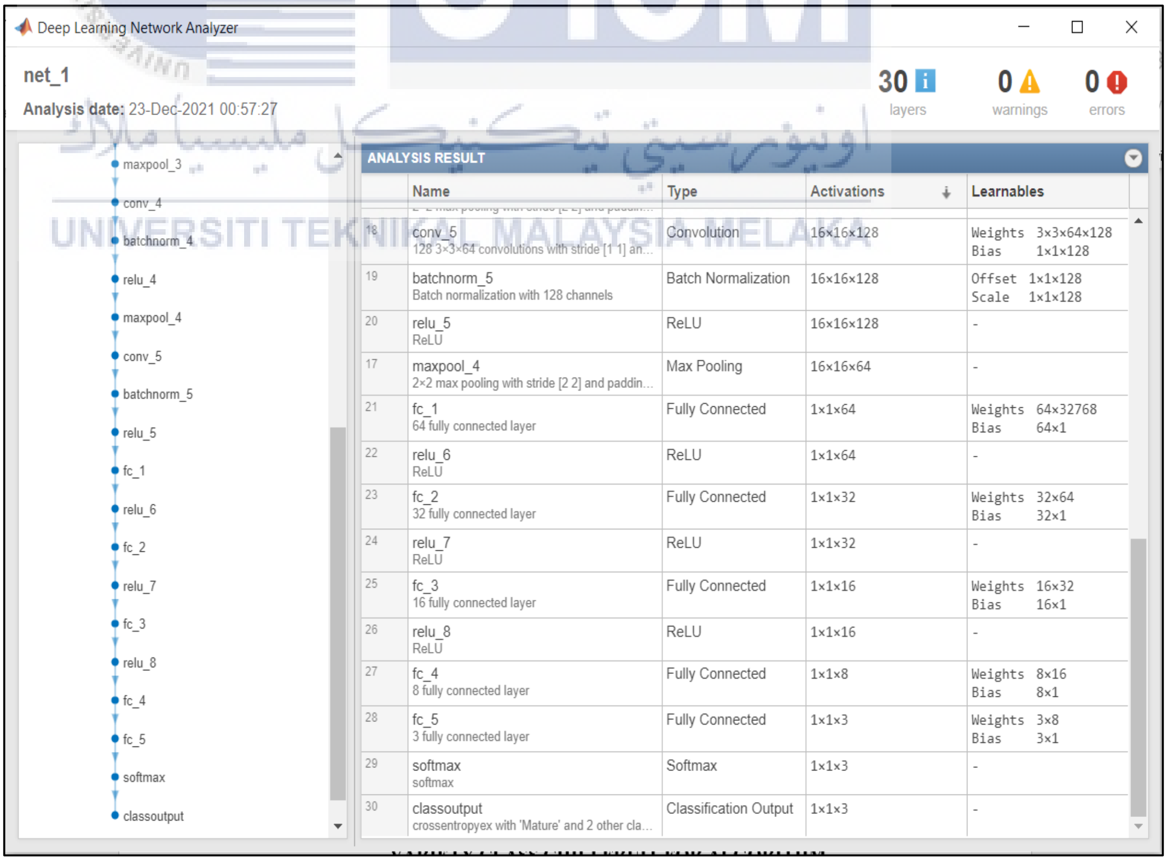
net_1

Analysis date: 23-Dec-2021 00:57:27

30 layers 0 warnings 0 errors

ANALYSIS RESULT

	Name	Type	Activations	Learnables
28	fc_5 3 fully connected layer	Fully Connected	1×1×3	Weights 3×8 Bias 3×1
29	softmax softmax	Softmax	1×1×3	-
30	classoutput crossentropyex with 'Mature' and 2 other cla...	Classification Output	1×1×3	-
27	fc_4 8 fully connected layer	Fully Connected	1×1×8	Weights 8×16 Bias 8×1
25	fc_3 16 fully connected layer	Fully Connected	1×1×16	Weights 16×32 Bias 16×1
26	relu_8 ReLU	ReLU	1×1×16	-
23	fc_2 32 fully connected layer	Fully Connected	1×1×32	Weights 32×64 Bias 32×1
24	relu_7 ReLU	ReLU	1×1×32	-
21	fc_1 64 fully connected layer	Fully Connected	1×1×64	Weights 64×32768 Bias 64×1
22	relu_6 ReLU	ReLU	1×1×64	-
17	maxpool_4 2×2 max pooling with stride [2 2] and paddin...	Max Pooling	16×16×64	-
13	maxpool_3 2×2 max pooling with stride [2 2] and paddin...	Max Pooling	32×32×32	-
18	conv_5 128 3×3×64 convolutions with stride [1 1] an...	Convolution	16×16×128	Weights 3×3×64×128 Bias 1×1×128
19	batchnorm_5 Batch normalization with 128 channels	Batch Normalization	16×16×128	Offset 1×1×128 Scale 1×1×128



net_1

Analysis date: 23-Dec-2021 00:57:27

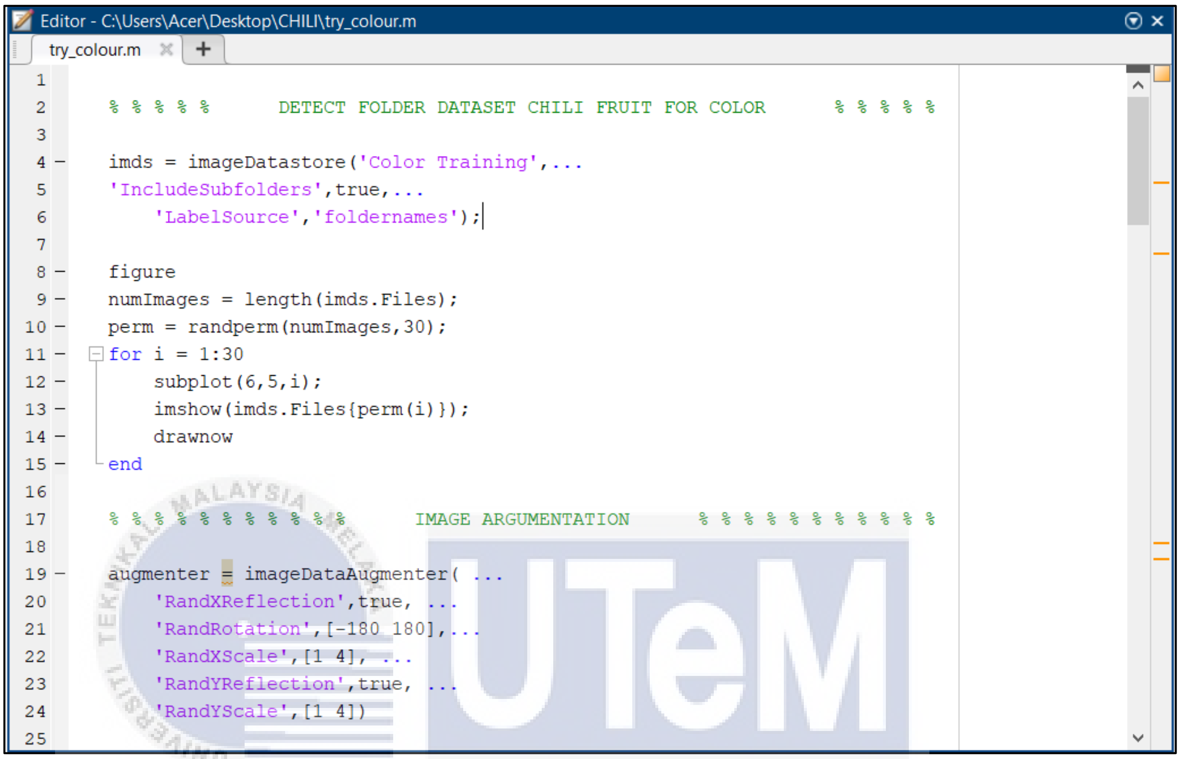
30 layers 0 warnings 0 errors

ANALYSIS RESULT

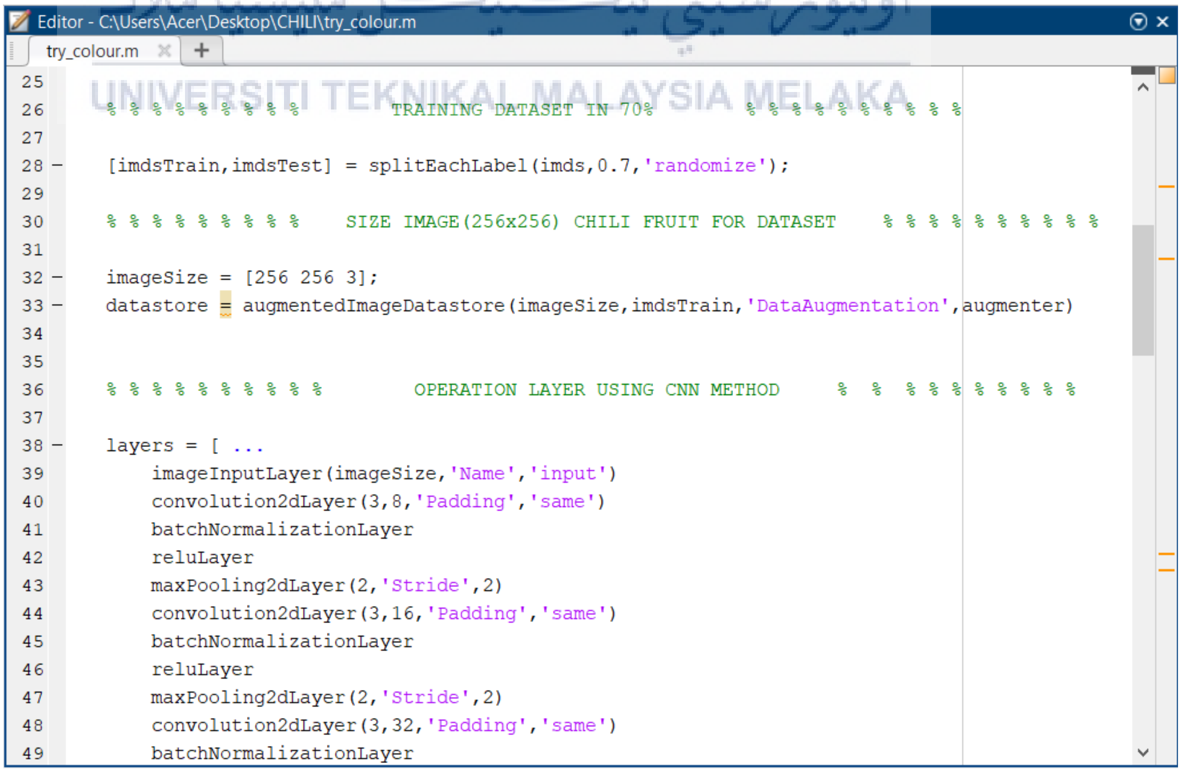
	Name	Type	Activations	Learnables
18	conv_5 128 3×3×64 convolutions with stride [1 1] an...	Convolution	16×16×128	Weights 3×3×64×128 Bias 1×1×128
19	batchnorm_5 Batch normalization with 128 channels	Batch Normalization	16×16×128	Offset 1×1×128 Scale 1×1×128
20	relu_5 ReLU	ReLU	16×16×128	-
17	maxpool_4 2×2 max pooling with stride [2 2] and paddin...	Max Pooling	16×16×64	-
21	fc_1 64 fully connected layer	Fully Connected	1×1×64	Weights 64×32768 Bias 64×1
22	relu_6 ReLU	ReLU	1×1×64	-
23	fc_2 32 fully connected layer	Fully Connected	1×1×32	Weights 32×64 Bias 32×1
24	relu_7 ReLU	ReLU	1×1×32	-
25	fc_3 16 fully connected layer	Fully Connected	1×1×16	Weights 16×32 Bias 16×1
26	relu_8 ReLU	ReLU	1×1×16	-
27	fc_4 8 fully connected layer	Fully Connected	1×1×8	Weights 8×16 Bias 8×1
28	fc_5 3 fully connected layer	Fully Connected	1×1×3	Weights 3×8 Bias 3×1
29	softmax softmax	Softmax	1×1×3	-
30	classoutput crossentropyex with 'Mature' and 2 other cla...	Classification Output	1×1×3	-

APPENDIX C

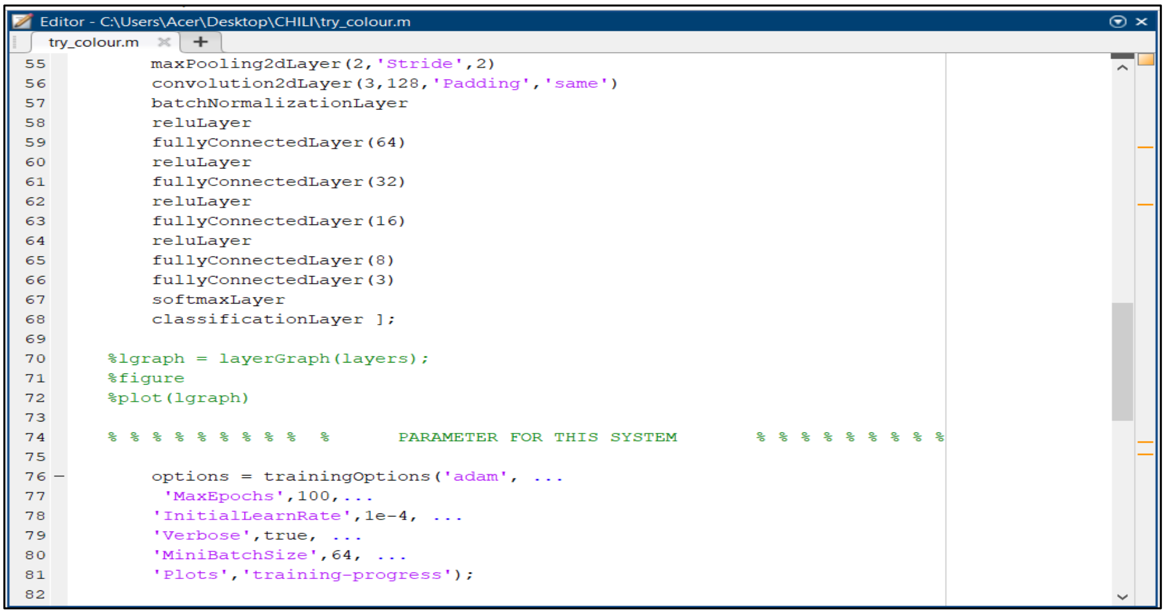
CODING FOR VARIETY CLASS CHILI FRUIT



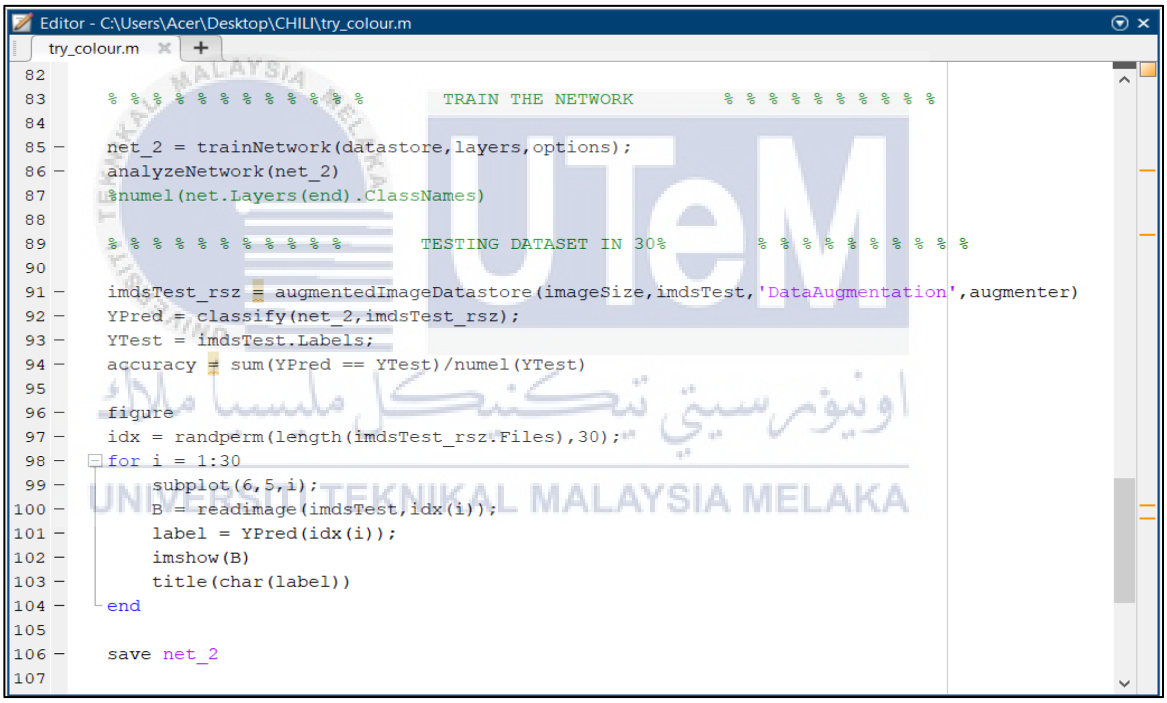
```
% % % % %      DETECT FOLDER DATASET CHILI FRUIT FOR COLOR      % % % % %

imds = imageDatastore('Color Training',...
'IncludeSubfolders',true,...
    'LabelSource','foldernames');

figure
numImages = length(imds.Files);
perm = randperm(numImages,30);
for i = 1:30
    subplot(6,5,i);
    imshow(imds.Files{perm(i)});
    drawnow
end

% % % % % % % % % % %      IMAGE ARGUMENTATION      % % % % % % % % % % %

augmenter = imageDataAugmenter( ...
    'RandXReflection',true, ...
    'RandRotation',[-180 180],...
    'RandXScale',[1 4], ...
    'RandYReflection',true, ...
    'RandYScale',[1 4])

```



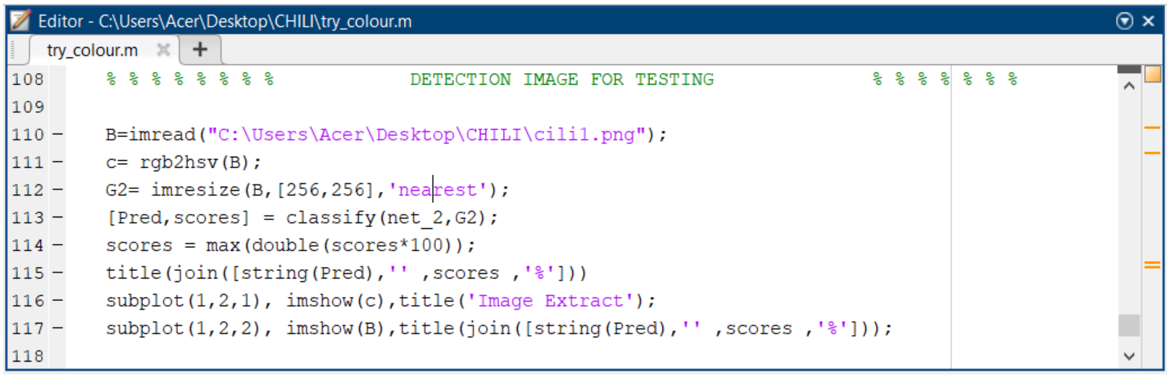
```

% % % % % % % % %      TRAINING DATASET IN 70%      % % % % % % % % % %

[imdsTrain,imdsTest] = splitEachLabel(imds,0.7,'randomize');

% % % % % % % % %    SIZE IMAGE(256x256) CHILI FRUIT FOR DATASET    % % % % % % % % % %

imageSize = [256 256 3];
datastore = augmentedImageDatastore(imageSize,imdsTrain,'DataAugmentation',augmenter)


% % % % % % % % % %      OPERATION LAYER USING CNN METHOD     %  %  % % % % % % % %

layers = [ ...
    imageInputLayer(imageSize,'Name','input')
    convolution2dLayer(3,8,'Padding','same')
    batchNormalizationLayer
    reluLayer
    maxPooling2dLayer(2,'Stride',2)
    convolution2dLayer(3,16,'Padding','same')
    batchNormalizationLayer
    reluLayer
    maxPooling2dLayer(2,'Stride',2)
    convolution2dLayer(3,32,'Padding','same')
    batchNormalizationLayer
```



```
        maxPooling2dLayer(2,'Stride',2)
        convolution2dLayer(3,128,'Padding','same')
        batchNormalizationLayer
        reluLayer
        fullyConnectedLayer(64)
        reluLayer
        fullyConnectedLayer(32)
        reluLayer
        fullyConnectedLayer(16)
        reluLayer
        fullyConnectedLayer(8)
        fullyConnectedLayer(3)
        softmaxLayer
        classificationLayer ];

    %lgraph = layerGraph(layers);
    %figure
    %plot(lgraph)

    % % % % % % % % %  %      PARAMETER FOR THIS SYSTEM       % % % % % % % % %

        options = trainingOptions('adam', ...
         'MaxEpochs',100,...
        'InitialLearnRate',1e-4, ...
        'Verbose',true, ...
        'MiniBatchSize',64, ...
        'Plots','training-progress');

```



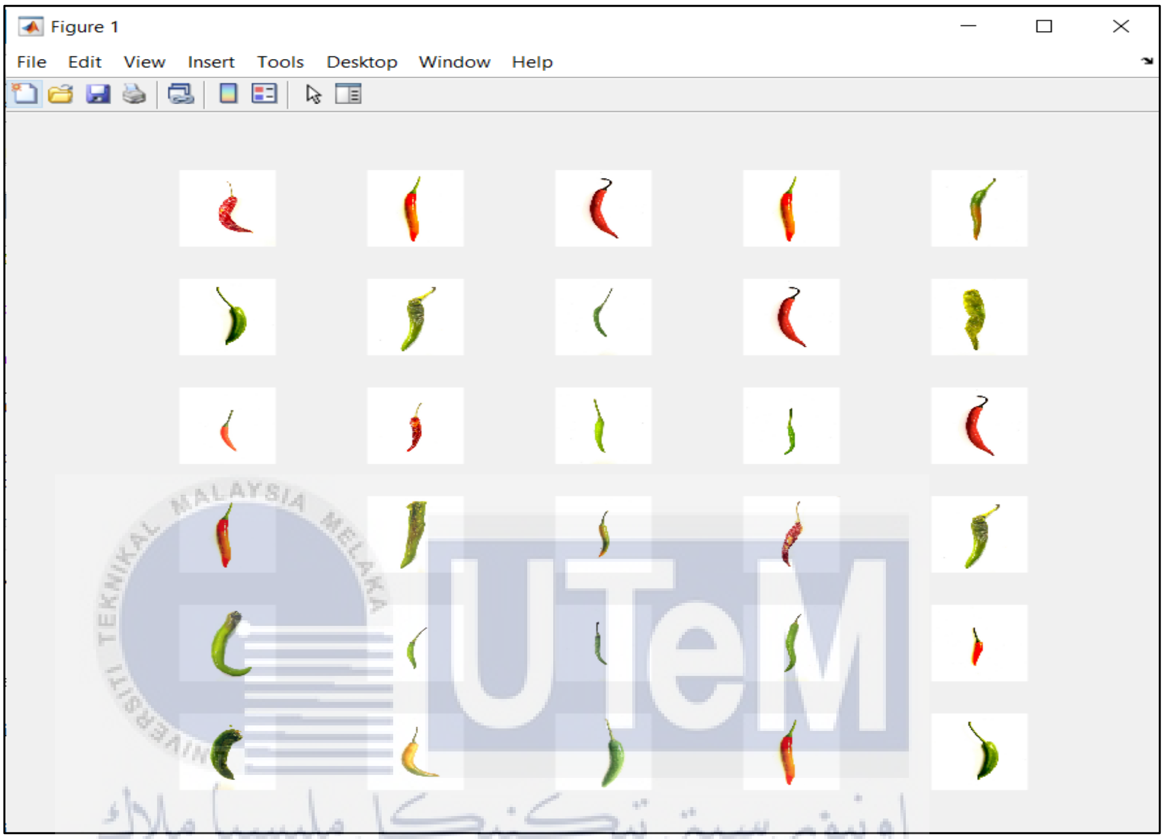
```

    % % % % % % % % % % % %      TRAIN THE NETWORK       % % % % % % % % % %

    net_2 = trainNetwork(datastore,layers,options);
    analyzeNetwork(net_2)
    %numel(net.Layers(end).ClassNames)

    % % % % % % % % % % %       TESTING DATASET IN 30%       % % % % % % % % % %

    imdsTest_rsz = augmentedImageDatastore(imageSize,imdsTest,'DataAugmentation',augmenter)
    YPred = classify(net_2,imdsTest_rsz);
    YTest = imdsTest.Labels;
    accuracy = sum(YPred == YTest)/numel(YTest)

    figure
    idx = randperm(length(imdsTest_rsz.Files),30);
    for i = 1:30
        subplot(6,5,i);
        B = readimage(imdsTest,idx(i));
        label = YPred(idx(i));
        imshow(B)
        title(char(label))
    end

    save net_2

```

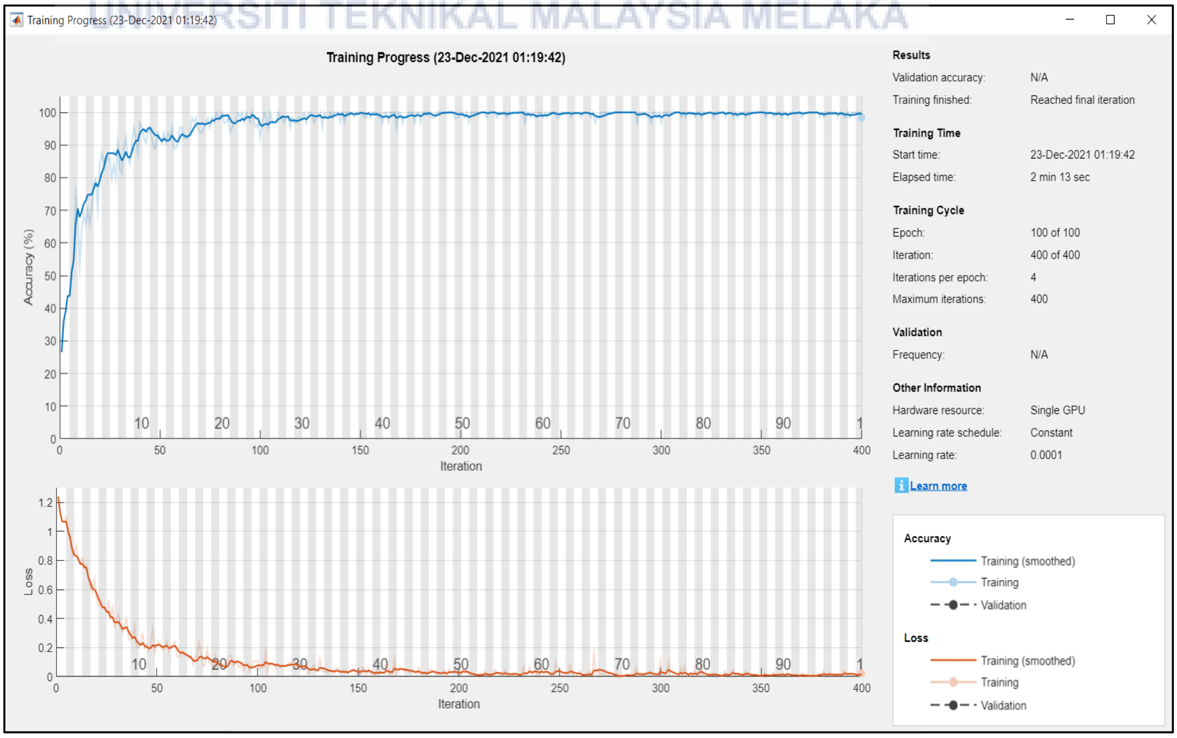


```
    % % % % % % % %          DETECTION IMAGE FOR TESTING          % % % % % % %

    B=imread("C:\Users\Acer\Desktop\CHILI\cili1.png");
    c= rgb2hsv(B);
    G2= imresize(B,[256,256],'nearest');
    [Pred,scores] = classify(net_2,G2);
    scores = max(double(scores*100));
    title(join([string(Pred),'' ,scores ,'%']))
    subplot(1,2,1), imshow(c),title('Image Extract');
    subplot(1,2,2), imshow(B),title(join([string(Pred),'' ,scores ,'%']));

```

APPENDIX D

RUN CODING FOR VARIETY CLASS





Training Progress (23-Dec-2021 01:19:42)

Results
Validation accuracy: N/A
Training finished: Reached final iteration

Training Time
Start time: 23-Dec-2021 01:19:42
Elapsed time: 2 min 13 sec

Training Cycle
Epoch: 100 of 100
Iteration: 400 of 400
Iterations per epoch: 4
Maximum iterations: 400

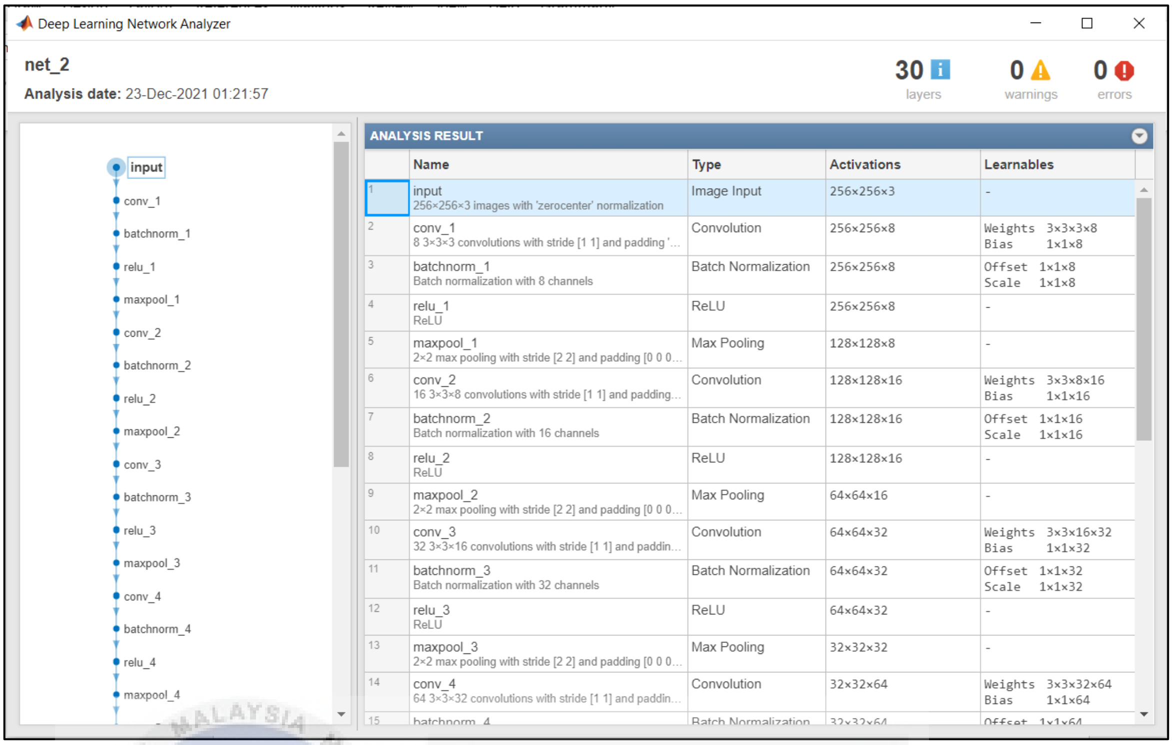
Validation
Frequency: N/A

Other Information
Hardware resource: Single GPU
Learning rate schedule: Constant
Learning rate: 0.0001

Learn more

Accuracy
Training (smoothed)
Training
Validation

Loss
Training (smoothed)
Training
Validation

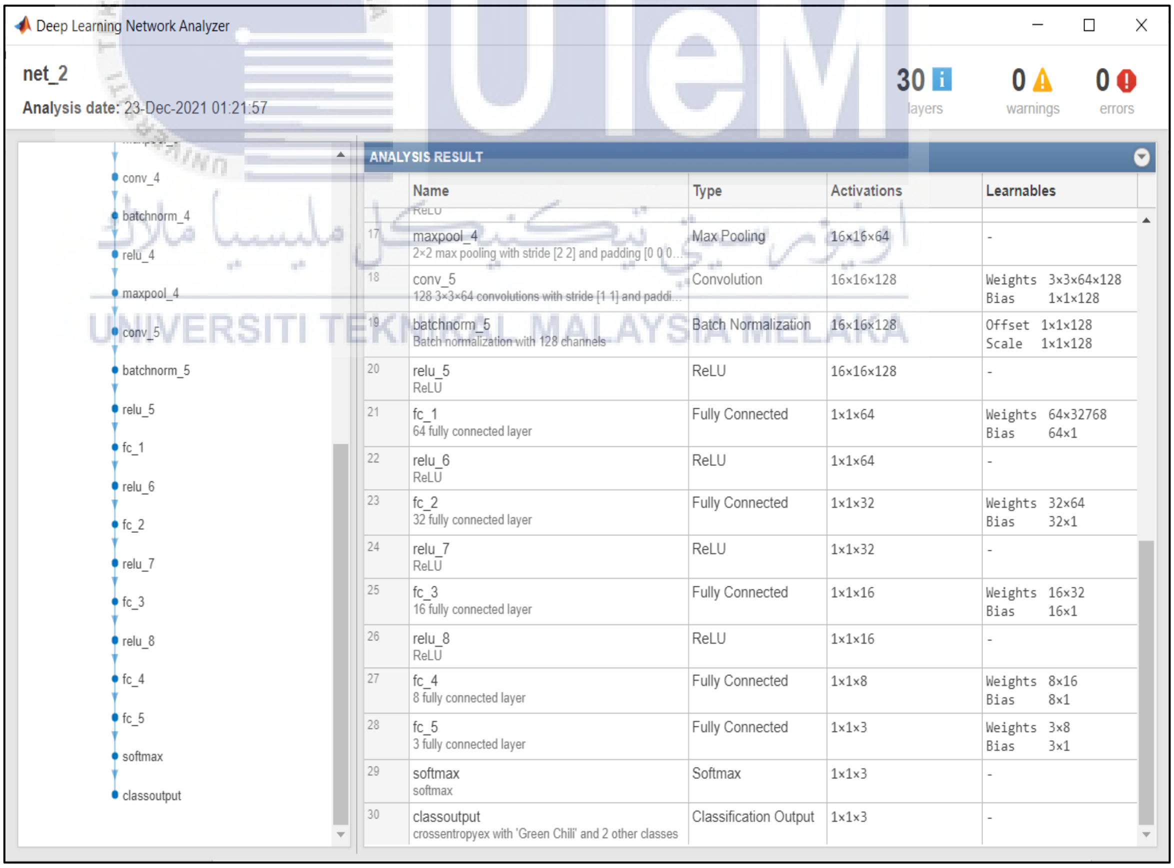


net_2
Analysis date: 23-Dec-2021 01:21:57

30 layers 0 warnings 0 errors

ANALYSIS RESULT

	Name	Type	Activations	Learnables
1	input 256×256×3 images with 'zerocenter' normalization	Image Input	256×256×3	-
2	conv_1 8 3×3×3 convolutions with stride [1 1] and padding '...	Convolution	256×256×8	Weights 3×3×3×8 Bias 1×1×8
3	batchnorm_1 Batch normalization with 8 channels	Batch Normalization	256×256×8	Offset 1×1×8 Scale 1×1×8
4	relu_1 ReLU	ReLU	256×256×8	-
5	maxpool_1 2×2 max pooling with stride [2 2] and padding [0 0 0...	Max Pooling	128×128×8	-
6	conv_2 16 3×3×8 convolutions with stride [1 1] and padding...	Convolution	128×128×16	Weights 3×3×8×16 Bias 1×1×16
7	batchnorm_2 Batch normalization with 16 channels	Batch Normalization	128×128×16	Offset 1×1×16 Scale 1×1×16
8	relu_2 ReLU	ReLU	128×128×16	-
9	maxpool_2 2×2 max pooling with stride [2 2] and padding [0 0 0...	Max Pooling	64×64×16	-
10	conv_3 32 3×3×16 convolutions with stride [1 1] and paddin...	Convolution	64×64×32	Weights 3×3×16×32 Bias 1×1×32
11	batchnorm_3 Batch normalization with 32 channels	Batch Normalization	64×64×32	Offset 1×1×32 Scale 1×1×32
12	relu_3 ReLU	ReLU	64×64×32	-
13	maxpool_3 2×2 max pooling with stride [2 2] and padding [0 0 0...	Max Pooling	32×32×32	-
14	conv_4 64 3×3×32 convolutions with stride [1 1] and paddin...	Convolution	32×32×64	Weights 3×3×32×64 Bias 1×1×64
15	batchnorm_4	Batch Normalization	32×32×64	Offset 1×1×64



net_2
Analysis date: 23-Dec-2021 01:21:57

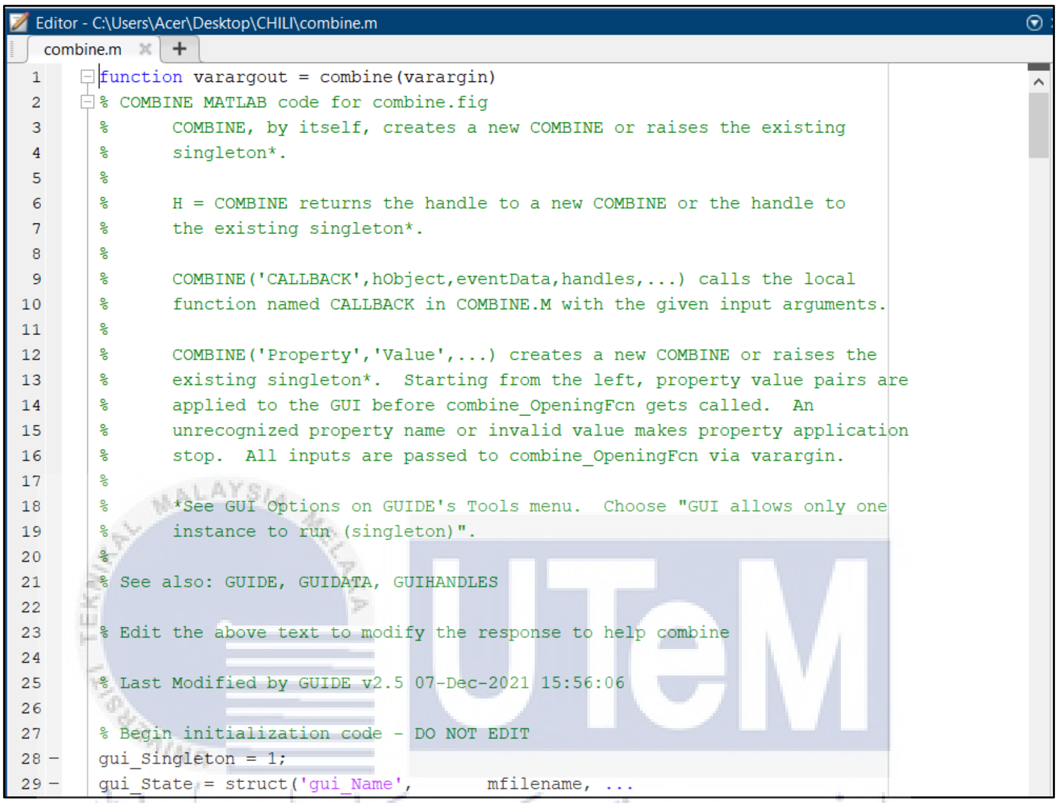
30 layers 0 warnings 0 errors

ANALYSIS RESULT

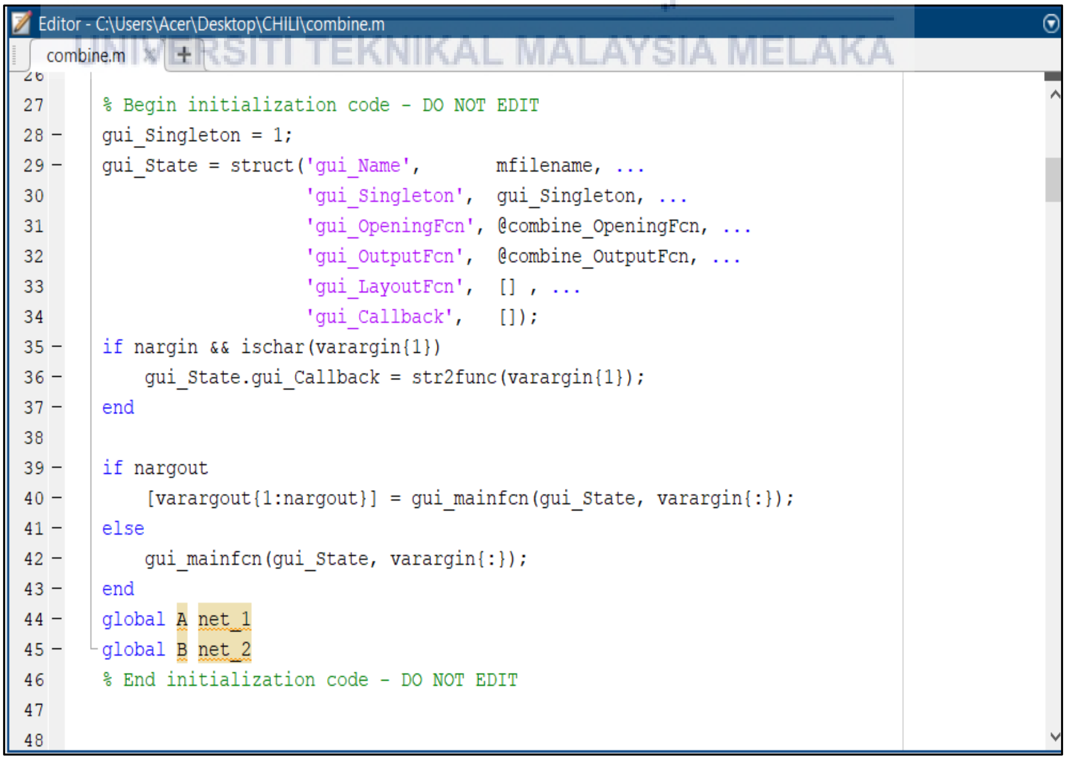
	Name	Type	Activations	Learnables
	ReLU			
17	maxpool_4 2×2 max pooling with stride [2 2] and padding [0 0 0...	Max Pooling	16×16×64	-
18	conv_5 128 3×3×64 convolutions with stride [1 1] and paddi...	Convolution	16×16×128	Weights 3×3×64×128 Bias 1×1×128
19	batchnorm_5 Batch normalization with 128 channels	Batch Normalization	16×16×128	Offset 1×1×128 Scale 1×1×128
20	relu_5 ReLU	ReLU	16×16×128	-
21	fc_1 64 fully connected layer	Fully Connected	1×1×64	Weights 64×32768 Bias 64×1
22	relu_6 ReLU	ReLU	1×1×64	-
23	fc_2 32 fully connected layer	Fully Connected	1×1×32	Weights 32×64 Bias 32×1
24	relu_7 ReLU	ReLU	1×1×32	-
25	fc_3 16 fully connected layer	Fully Connected	1×1×16	Weights 16×32 Bias 16×1
26	relu_8 ReLU	ReLU	1×1×16	-
27	fc_4 8 fully connected layer	Fully Connected	1×1×8	Weights 8×16 Bias 8×1
28	fc_5 3 fully connected layer	Fully Connected	1×1×3	Weights 3×8 Bias 3×1
29	softmax softmax	Softmax	1×1×3	-
30	classoutput crossentropyex with 'Green Chili' and 2 other classes	Classification Output	1×1×3	-

APPENDIX E

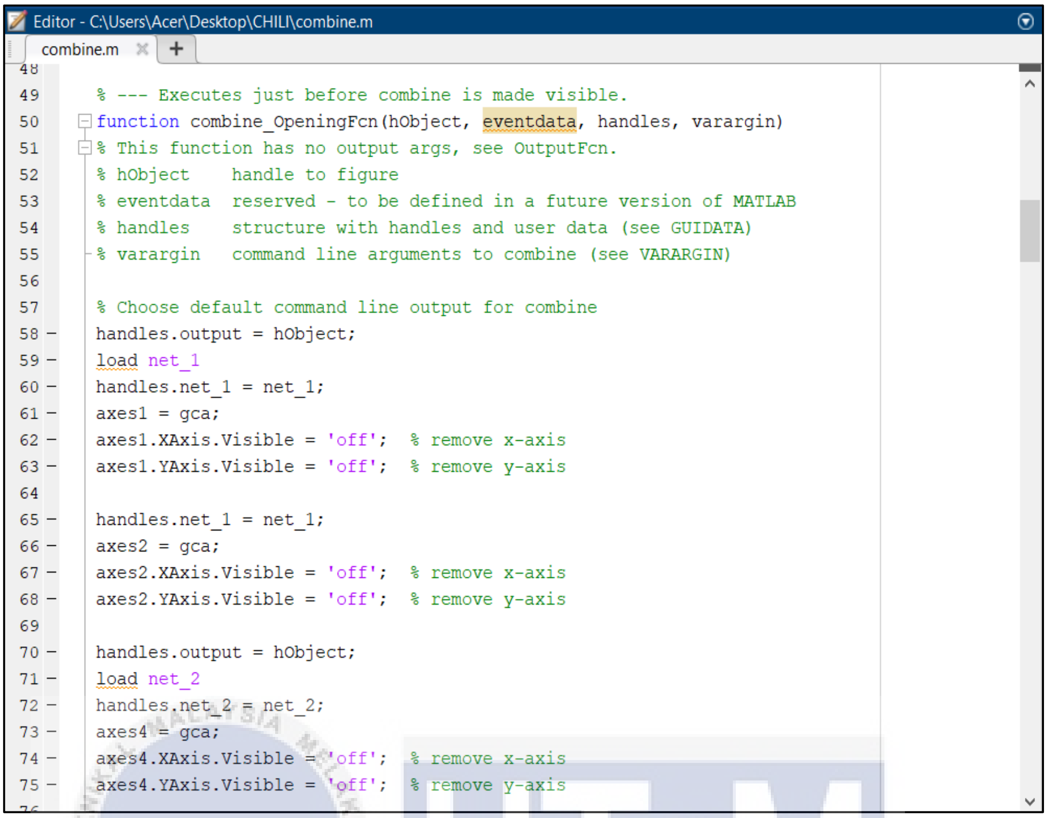
GUI SYSTEM CODING



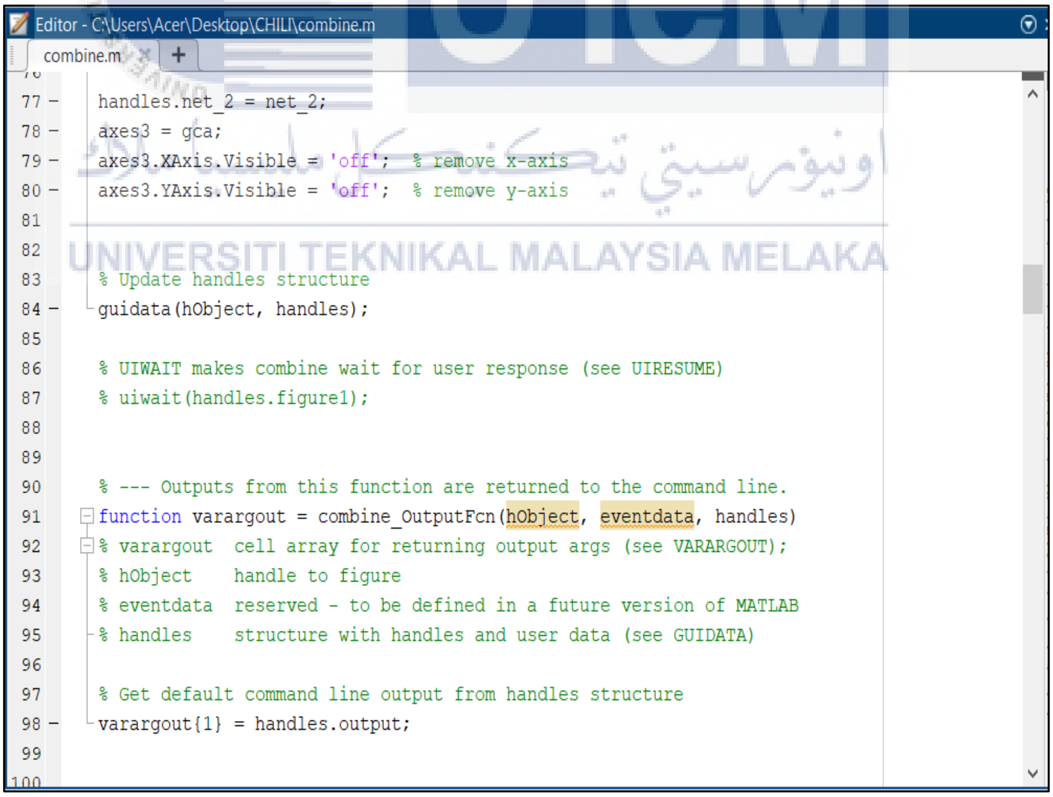
```
function varargout = combine(varargin)
% COMBINE MATLAB code for combine.fig
%      COMBINE, by itself, creates a new COMBINE or raises the existing
%      singleton*.
%
%      H = COMBINE returns the handle to a new COMBINE or the handle to
%      the existing singleton*.
%
%      COMBINE('CALLBACK',hObject,eventData,handles,...) calls the local
%      function named CALLBACK in COMBINE.M with the given input arguments.
%
%      COMBINE('Property','Value',...) creates a new COMBINE or raises the
%      existing singleton*.  Starting from the left, property value pairs are
%      applied to the GUI before combine_OpeningFcn gets called.  An
%      unrecognized property name or invalid value makes property application
%      stop.  All inputs are passed to combine_OpeningFcn via varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help combine

% Last Modified by GUIDE v2.5 07-Dec-2021 15:56:06

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
```



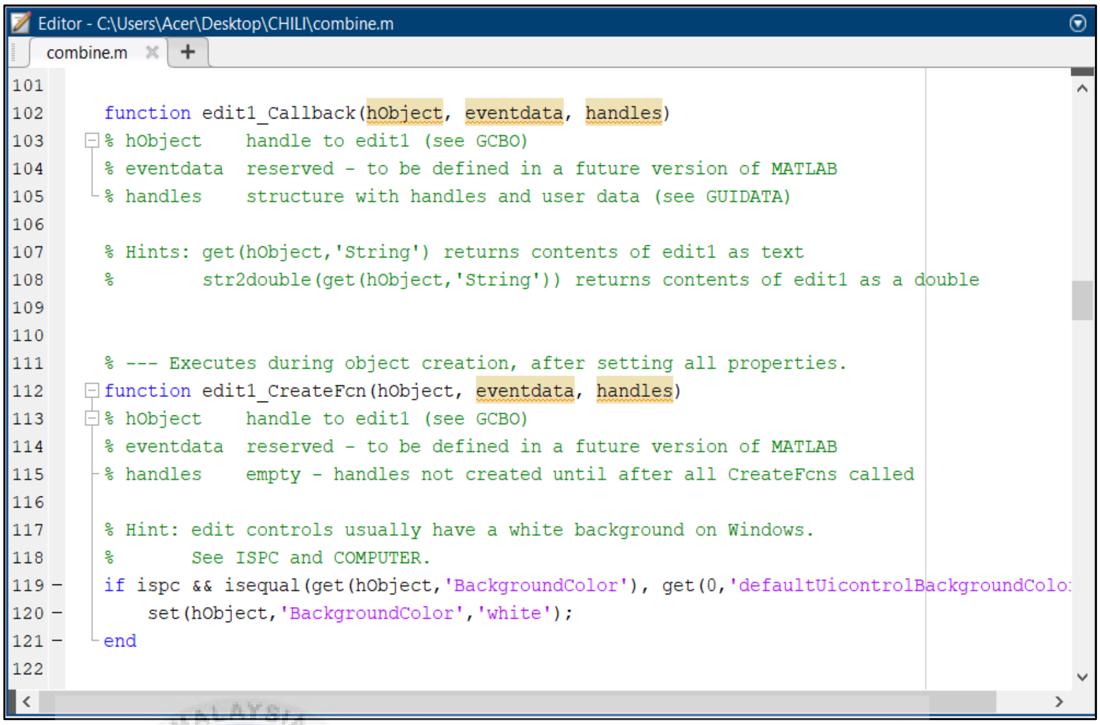
```
% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @combine_OpeningFcn, ...
                   'gui_OutputFcn',  @combine_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
global A net_1
global B net_2
% End initialization code - DO NOT EDIT
```



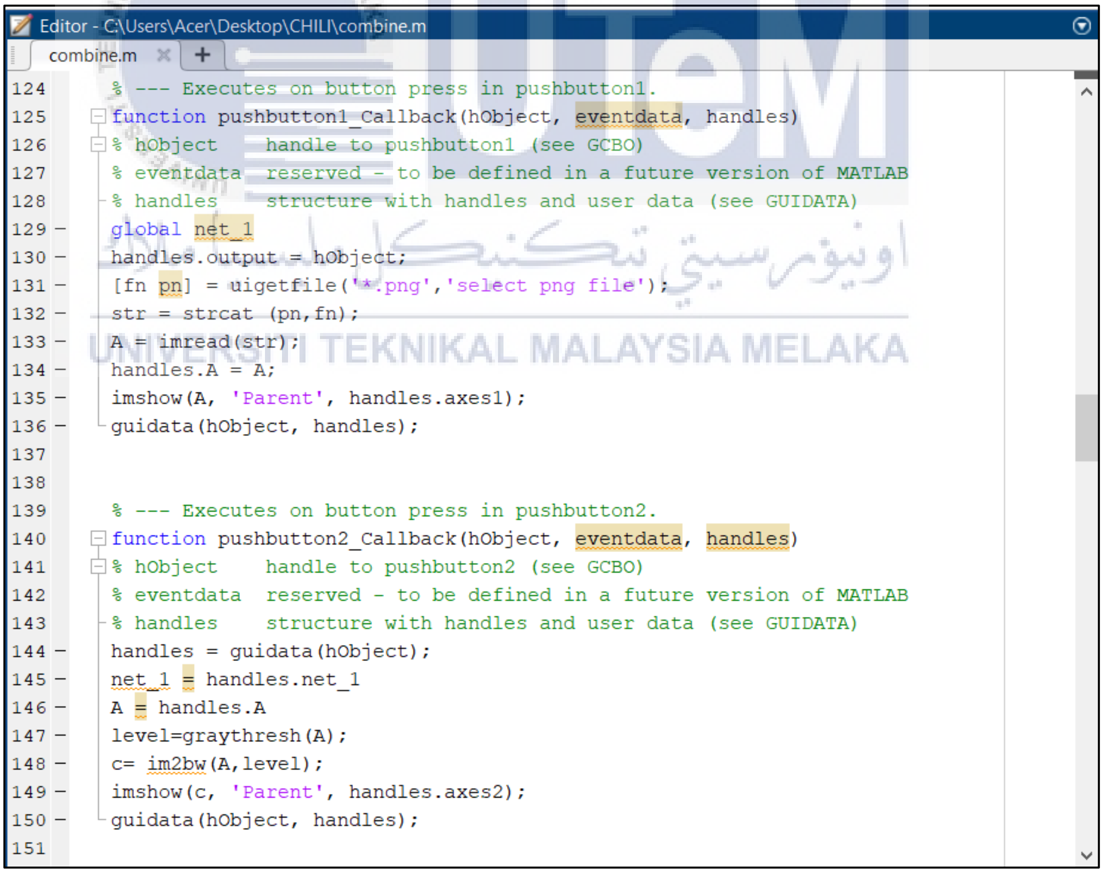
```
% --- Executes just before combine is made visible.
function combine_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to combine (see VARARGIN)

% Choose default command line output for combine
handles.output = hObject;
load net_1
handles.net_1 = net_1;
axes1 = gca;
axes1.XAxis.Visible = 'off';  % remove x-axis
axes1.YAxis.Visible = 'off';  % remove y-axis

handles.net_1 = net_1;
axes2 = gca;
axes2.XAxis.Visible = 'off';  % remove x-axis
axes2.YAxis.Visible = 'off';  % remove y-axis

handles.output = hObject;
load net_2
handles.net_2 = net_2;
axes4 = gca;
axes4.XAxis.Visible = 'off';  % remove x-axis
axes4.YAxis.Visible = 'off';  % remove y-axis
```



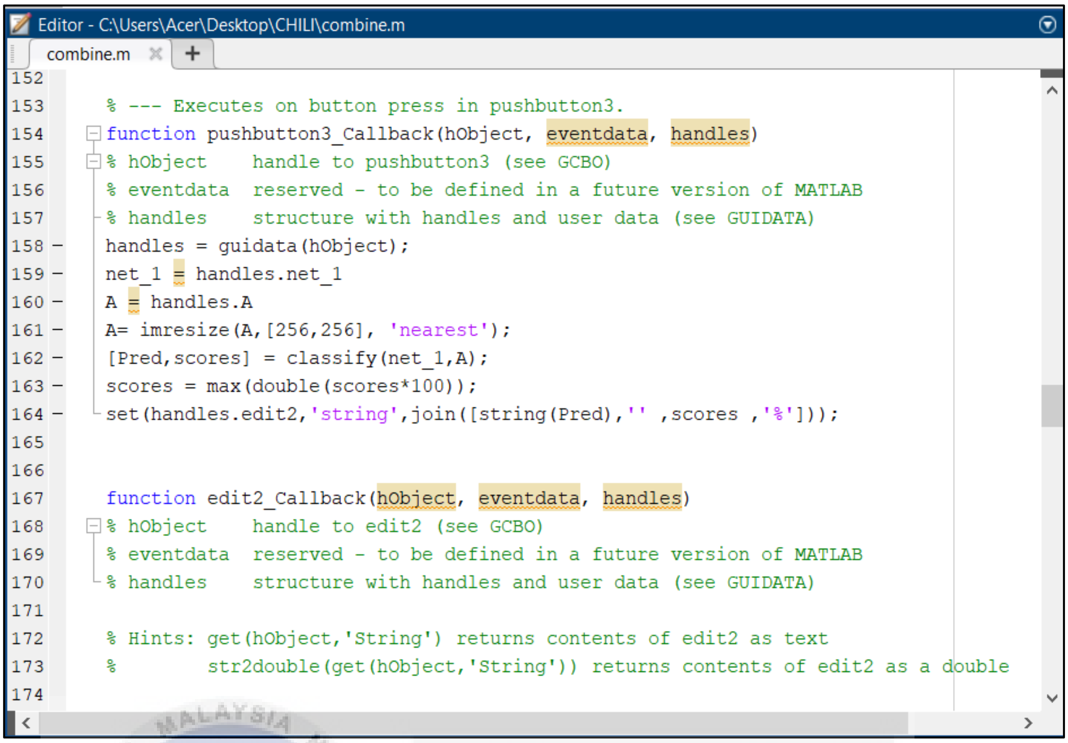
```
handles.net_2 = net_2;
axes3 = gca;
axes3.XAxis.Visible = 'off';  % remove x-axis
axes3.YAxis.Visible = 'off';  % remove y-axis


% Update handles structure
guidata(hObject, handles);

% UIWAIT makes combine wait for user response (see UIRESUME)
% uiwait(handles.figure1);


% --- Outputs from this function are returned to the command line.
function varargout = combine_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;
```



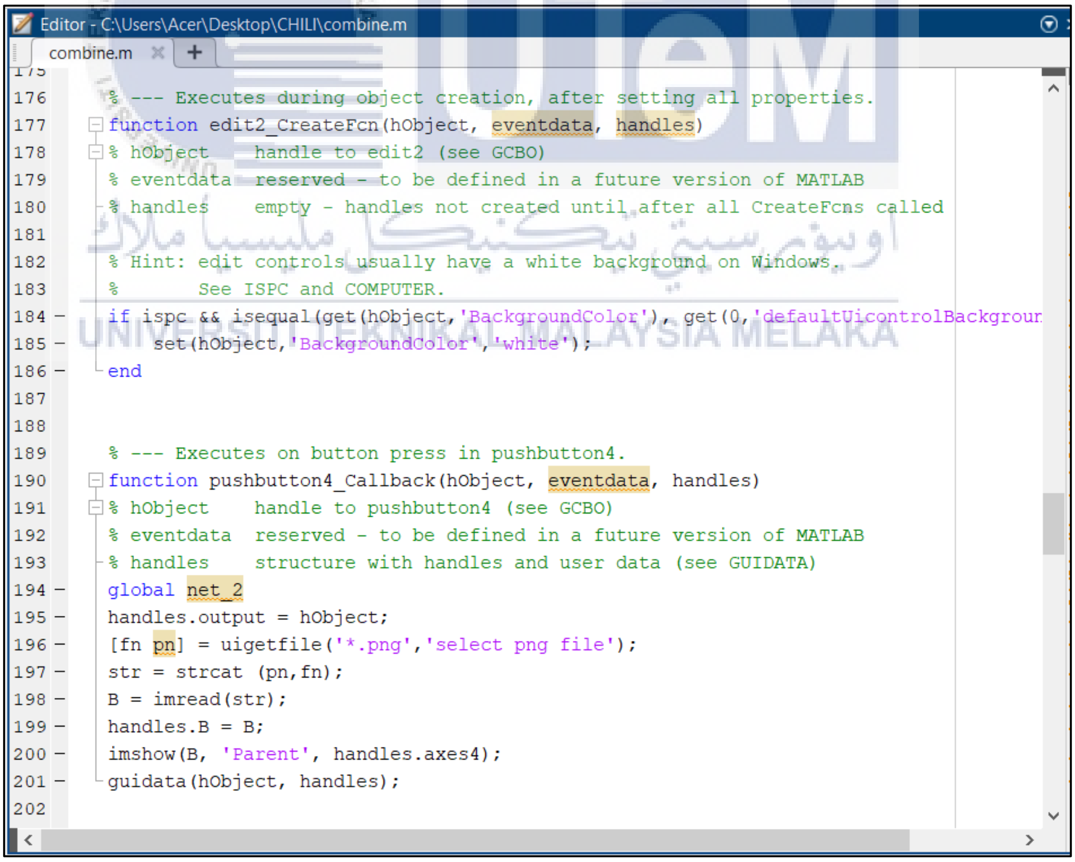
```
function edit1_Callback(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit1 as text
%        str2double(get(hObject,'String')) returns contents of edit1 as a double


% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColo
    set(hObject,'BackgroundColor','white');
end

```



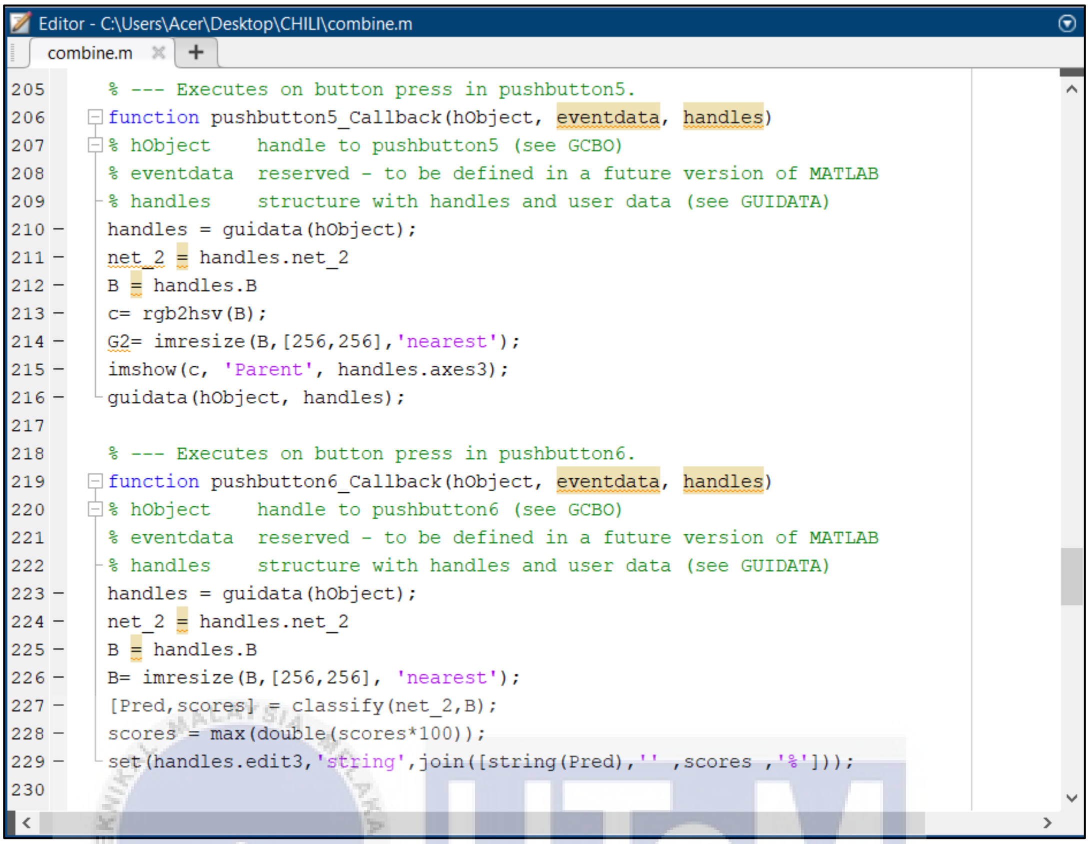
```
% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global net_1
handles.output = hObject;
[fn pn] = uigetfile('*.png','select png file');
str = strcat (pn,fn);
A = imread(str);
handles.A = A;
imshow(A, 'Parent', handles.axes1);
guidata(hObject, handles);


% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
handles = guidata(hObject);
net_1 = handles.net_1
A = handles.A
level=graythresh(A);
c= im2bw(A,level);
imshow(c, 'Parent', handles.axes2);
guidata(hObject, handles);

```



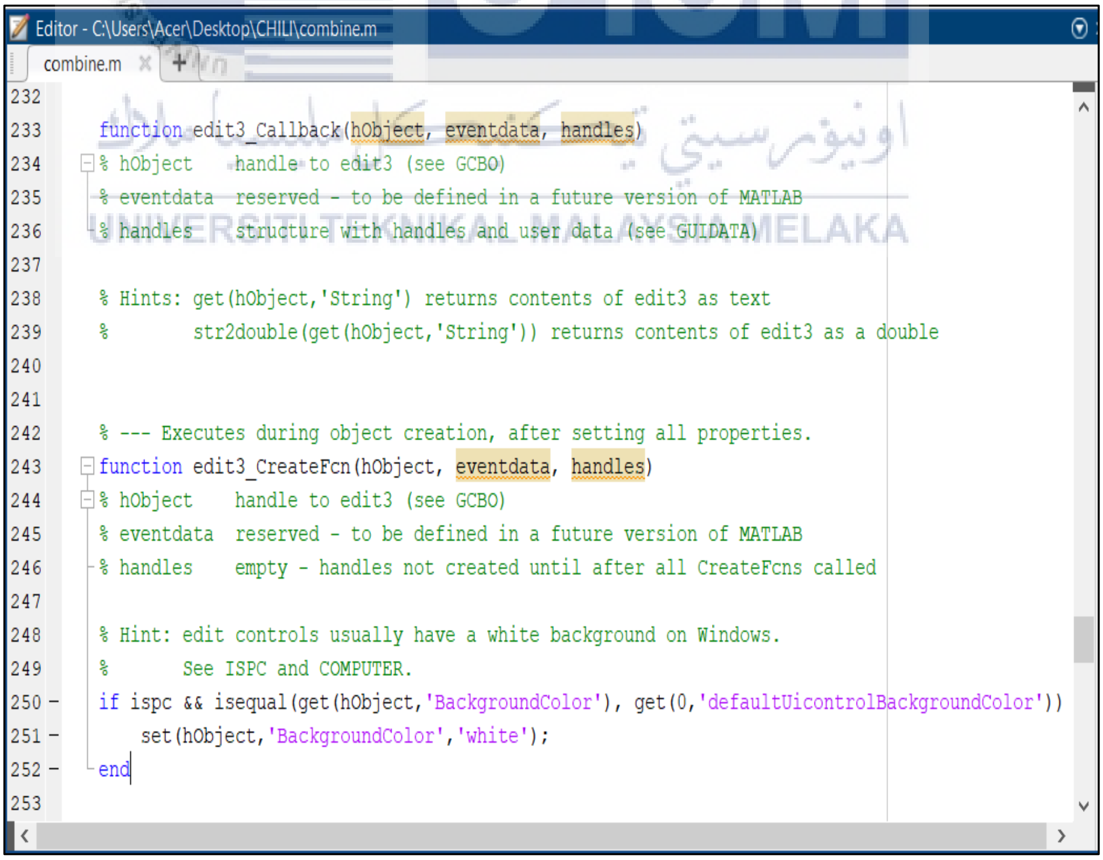
```
% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
handles = guidata(hObject);
net_1 = handles.net_1
A = handles.A
A= imresize(A,[256,256], 'nearest');
[Pred,scores] = classify(net_1,A);
scores = max(double(scores*100));
set(handles.edit2,'string',join([string(Pred),'' ,scores ,'%']));


function edit2_Callback(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit2 as text
%        str2double(get(hObject,'String')) returns contents of edit2 as a double
```



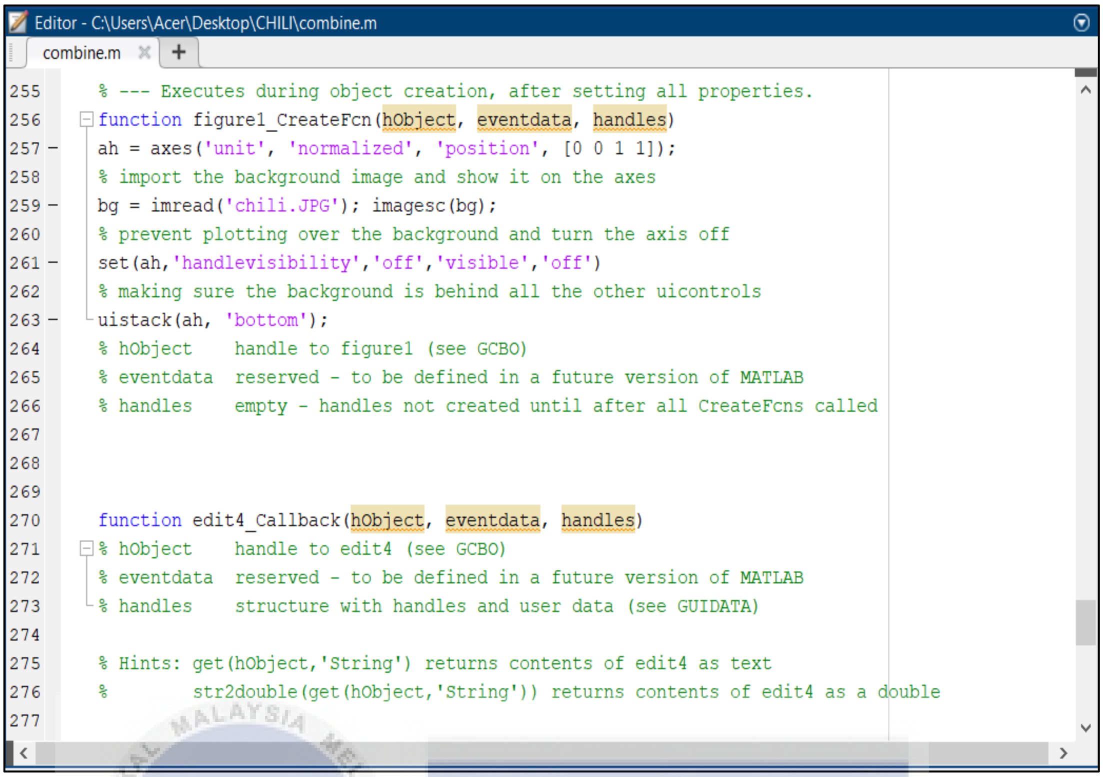
```
% --- Executes during object creation, after setting all properties.
function edit2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgrour
    set(hObject,'BackgroundColor','white');
end


% --- Executes on button press in pushbutton4.
function pushbutton4_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global net_2
handles.output = hObject;
[fn pn] = uigetfile('*.png','select png file');
str = strcat (pn,fn);
B = imread(str);
handles.B = B;
imshow(B, 'Parent', handles.axes4);
guidata(hObject, handles);
```



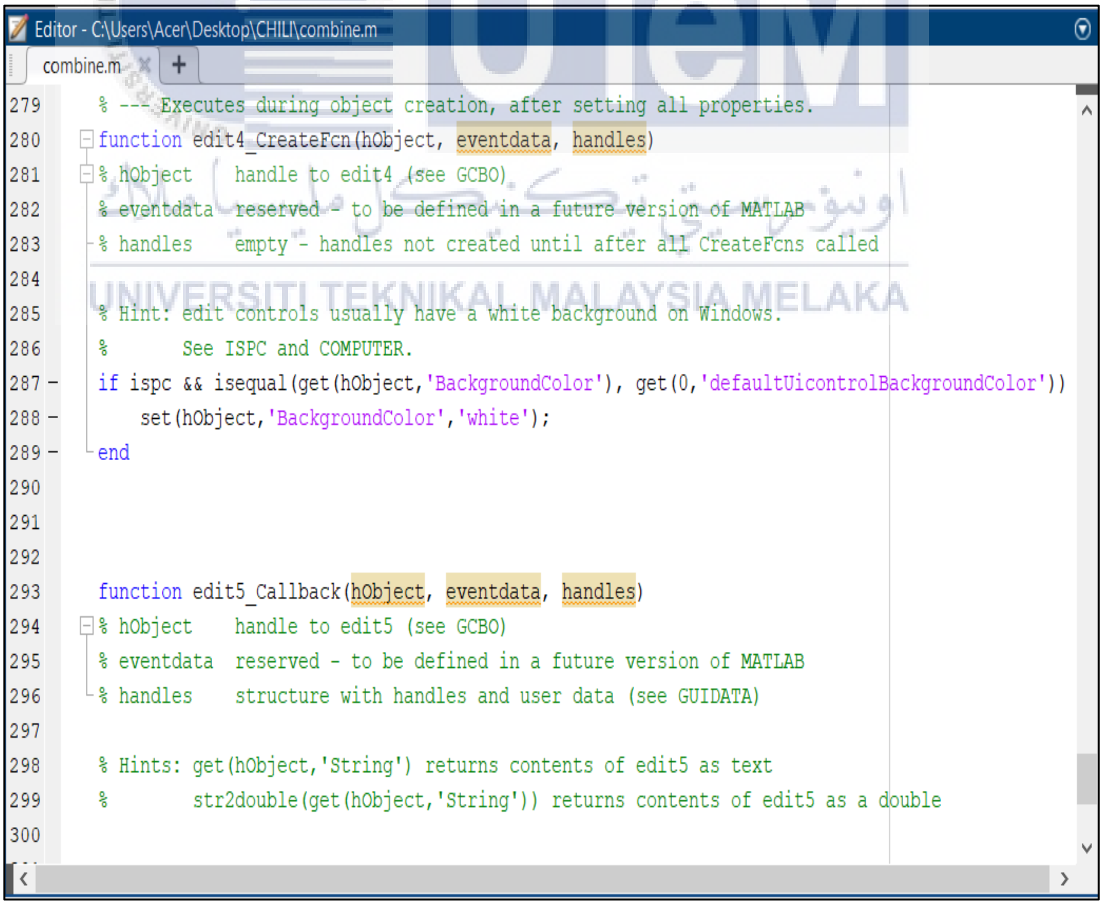
```
% --- Executes on button press in pushbutton5.
function pushbutton5_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
handles = guidata(hObject);
net_2 = handles.net_2
B = handles.B
c= rgb2hsv(B);
G2= imresize(B,[256,256],'nearest');
imshow(c, 'Parent', handles.axes3);
guidata(hObject, handles);

% --- Executes on button press in pushbutton6.
function pushbutton6_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton6 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
handles = guidata(hObject);
net_2 = handles.net_2
B = handles.B
B= imresize(B,[256,256], 'nearest');
[Pred,scores] = classify(net_2,B);
scores = max(double(scores*100));
set(handles.edit3,'string',join([string(Pred),'' ,scores ,'%']));

```



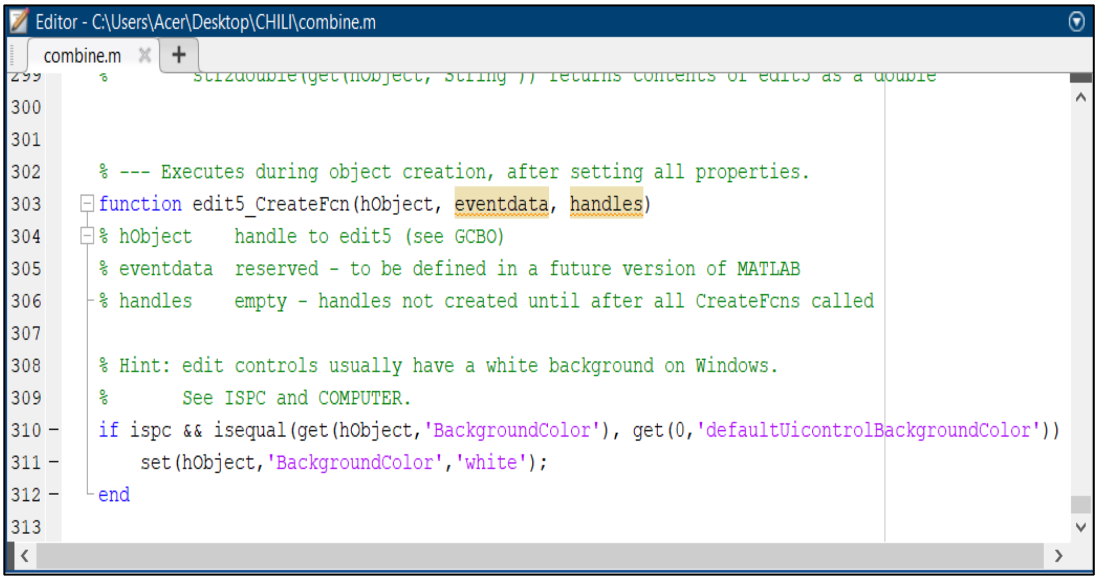
```

function edit3_Callback(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit3 as text
%        str2double(get(hObject,'String')) returns contents of edit3 as a double


% --- Executes during object creation, after setting all properties.
function edit3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```



```
% --- Executes during object creation, after setting all properties.
function figure1_CreateFcn(hObject, eventdata, handles)
ah = axes('unit', 'normalized', 'position', [0 0 1 1]);
% import the background image and show it on the axes
bg = imread('chili.JPG'); imagesc(bg);
% prevent plotting over the background and turn the axis off
set(ah,'handlevisibility','off','visible','off')
% making sure the background is behind all the other uicontrols
uistack(ah, 'bottom');
% hObject    handle to figure1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called



function edit4_Callback(hObject, eventdata, handles)
% hObject    handle to edit4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit4 as text
%        str2double(get(hObject,'String')) returns contents of edit4 as a double

```



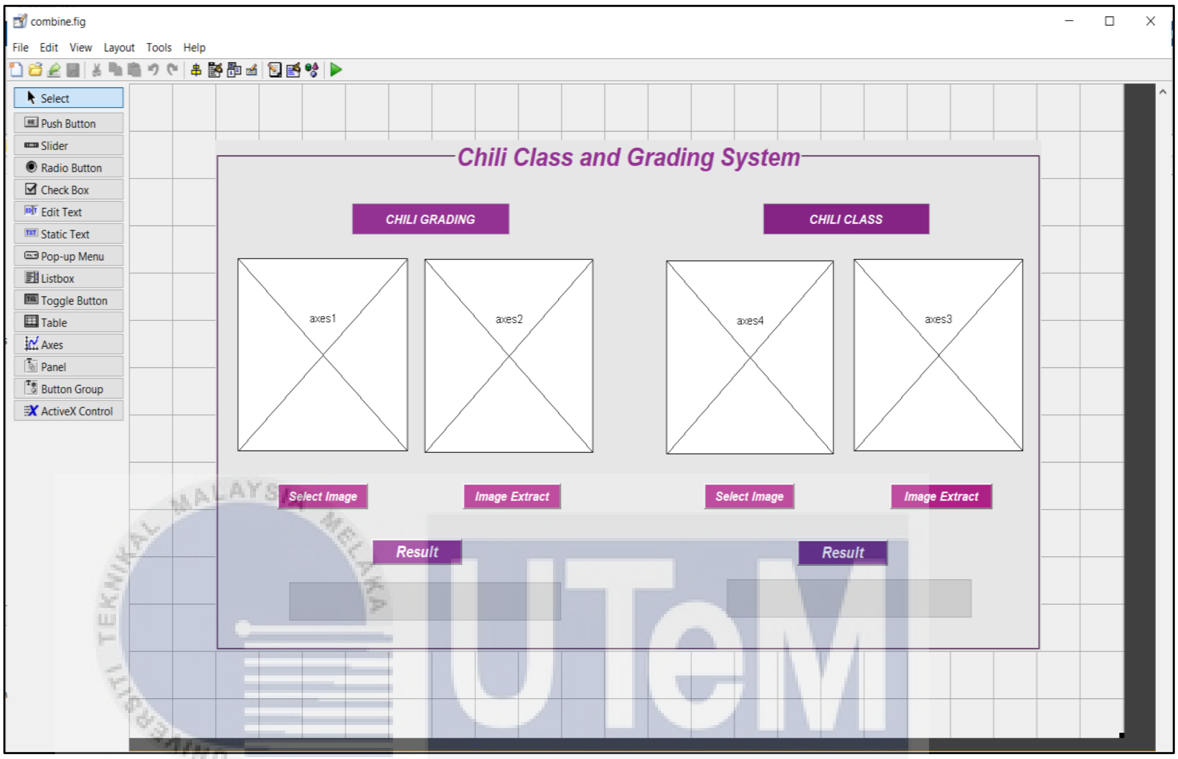
```
% --- Executes during object creation, after setting all properties.
function edit4_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit5_Callback(hObject, eventdata, handles)
% hObject    handle to edit5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit5 as text
%        str2double(get(hObject,'String')) returns contents of edit5 as a double

```



```
%        str2double(get(hObject,'String')) returns contents of edit5 as a double


% --- Executes during object creation, after setting all properties.
function edit5_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

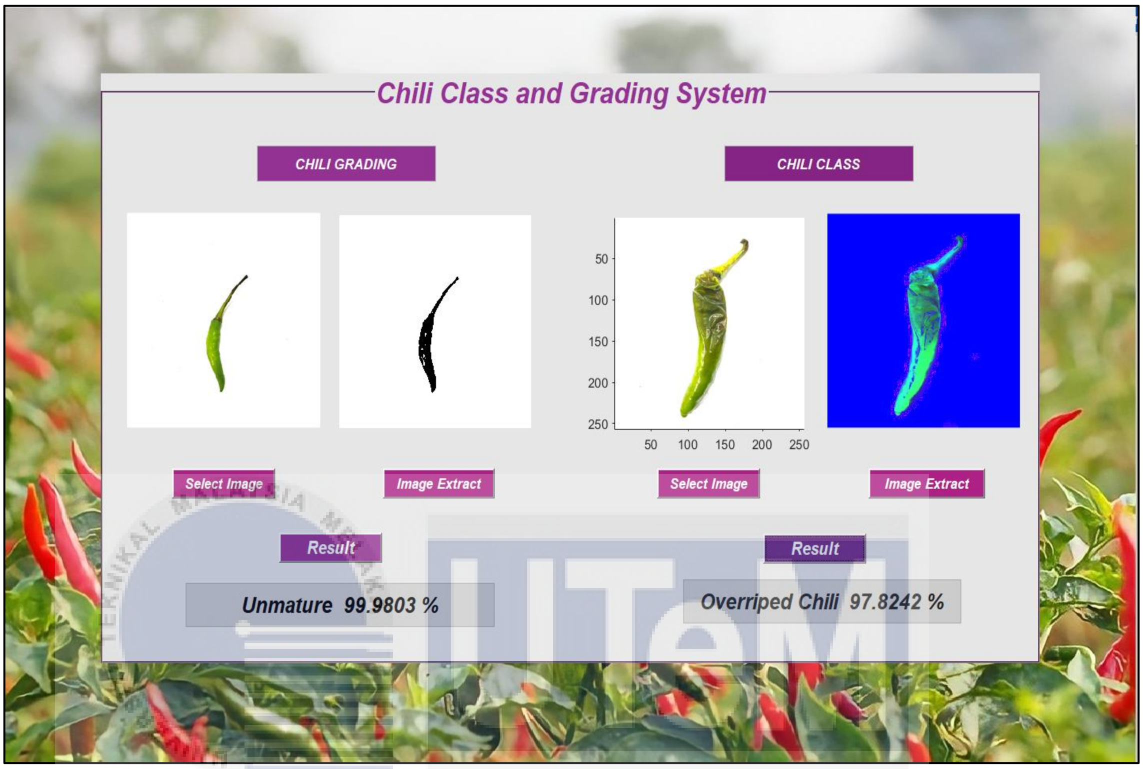
APPENDIX F

GUI SYSTEM DESIGN



APPENDIX G

GUI SYSTEM FOR GRADING AND CLASS CHILI FRUIT



Unmature 99.9803 %

Overriped Chili 97.8242 %

APPENDIX H

PROJECT PLANNING

B. PERANCANGAN PROJEK
PROJECT PLANNING (GANTT CHART)

Senaraikan aktiviti-aktiviti yang berkaitan bagi projek yang dicadangkan dan nyatakan jangka masa yang diperlukan bagi setiap aktiviti.
List all the relevant activities of the proposed project and mark the period of time that is needed for each of the activities.

	SEM II																			**SEM BREAK**					**SEM I**															
Aktiviti Projek *Project Activities*	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1.0 TITLE SELECTION											**SEMINAR PSM I**																											**SEMINAR PSM II**		
Choose PSM Tittle and Supervisor	X																																							
2.0 PROJECT PROPOSAL																																								
Make A Research and Understand Aim of Project	X																																							
Preparation of Proposal and Check by Supervisor		X	■																																					
Proposal Defense				X																																				
3.0 PSM 1																																								
Planning					X	■																																		

B. PERANCANGAN PROJEK
PROJECT PLANNING (GANTT CHART)

Senaraikan aktiviti-aktiviti yang berkaitan bagi projek yang dicadangkan dan nyatakan jangka masa yang diperlukan bagi setiap aktiviti.
List all the relevant activities of the proposed project and mark the period of time that is needed for each of the activities.

	SEM II																			SEM BREAK					SEM I															
Aktiviti Projek *Project Activities*	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Chapter 1 & 2: Introduction & Literature Review						X					SEMINAR PSM I																											SEMINAR PSM II		
Check with Supervisor							X																																	
Data Collection								X																																
Chapter 3: Methodology									X																															
Training & Testing for Maturity Category by Size									X	X																														
Check with Supervisor										X																														
Correction Slide Presentation and Draft Thesis PSM 1												X																												
Check with Supervisor														X																										
PSM 1 Presentation															X																									
Submit Logbook & Draft Thesis 1															X																									
2.0 PSM 2																																								
Continue the Training & Testing for Variety Class to its Color																						X																		

B. PERANCANGAN PROJEK
PROJECT PLANNING (GANTT CHART)

Senaraikan aktiviti-aktiviti yang berkaitan bagi projek yang dicadangkan dan nyatakan jangka masa yang diperlukan bagi setiap aktiviti.
List all the relevant activities of the proposed project and mark the period of time that is needed for each of the activities.

	SEM II																			**SEM BREAK**					**SEM I**															
Aktiviti Projek *Project Activities*	**1**	2	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	12	13	**14**	**15**	**16**	17	**18**	**19**	**20**	**21**	22	23	**24**	1	**2**	3	4	**5**	6	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**
Chapter 4: Result											**SEMINAR PSM I**														X	▨												**SEMINAR PSM II**		
Analyze the Various CNN Architecture																										X	▨	▨												
Check with Supervisor																													**X**	▨										
Correction Draft Thesis																															**X**	▨								
Design GUI System for the Grading and Class Chili Fruit																																**X**	▨							
Chapter 4: Discussion																																**X**	▨	▨						
Check with Supervisor																																		**X**	▨					
Chapter 5: Conclusion & Future Work																																				**X**				
Slide Preparation																																					**X**			
Final Thesis Submission to Panel																																							**X**	
Submission of Evaluation Marks to JK PSM																																							**X**	▨