Evolution in Electrical and Electronic Engineering Vol. 4 No. 1 (2023) 185-194

EEEE

Homepage: http://publisher.uthm.edu.my/periodicals/index.php/eeee
e-ISSN : 2756-8458

Grading Oil Palm Fruit Bunch using Convolution Neural Network

Hamdanzakirin Azman[1], Nor Surayahani Suriani[1]*

[1]Department of Electronic Engineering, Faculty of Electrical and Electronic Engineering,
Universiti Tun Onn Malaysia, 86400, Batu Pahat, Johor, Malaysia

*Corresponding Author Designation

DOI: https://doi.org/10.30880/eeee.2023.04.01.022
Received 15 January 2023; Accepted 10 April 2023; Available online 30 April 2023

Abstract: Elaeis guineensis is a typical oil palm fresh fruit bunches (FFB) in Malaysia that must be harvested at the optimum ripeness. In order to effectively assess the quality of oil palm fruit (FFB), non-contact image sensing technology can provide automatic and non -destructive detection fruit itself. Oil palm fruit is harvested in conditions that FFB oil palm should not gather due to the raw fruit that looks ripe due to an error from the human vision to recognize the best state for ripe fruit. The expected FFB are ripe bunches with a yellowish and reddish outer layer and a yellow-colored mesocarp. Thus, the proposed system can determine the quality directly via an android smartphone to speed up the recognition of oil palm FFB. Convolution Neural Network (CNN) system will use to classify types of ripeness grading of the oil palm fruit via android smartphone. The overall results for 8458 total number of images for four classes of oil palm fruit bunch FFB (Overripe, Ripe, Underripe, and Unripe) is 93.19% accuracy using Anaconda Jupyter Notebook and Google Colab Pro platform. The model is deployed into the Android Studio successfully by using TensorFlow Lite to build an android application. The android application detection accuracy was 88.79% for the Samsung Galaxy S22 (SM-S901E) model with 1050ms inference time and 91% for the Samsung Galaxy A30 (SM-A305F) model with 1140ms inference time. This model approach can assist workers to determine the maturity level of oil palm fruit bunch before making a decision.

Keywords: Convolutional Neural Network, Classification, oil palm fruit bunch, Tensorflow, Android Studio, TensorFlow Lite.

1. Introduction

On 4 January 2021, there was a significant increase in crude palm oil prices, rising from RM 3903.00 per ton to RM 6602.50 per ton on 30 March 2022 [1]. This highlights the importance of harvesting oil palm fruit in the best condition to produce high-quality palm oil.

**Corresponding author: nsuraya@uthm.edu.my*

Determining the maturity of oil palm fruit before harvest requires a high level of expertise, and human grading can be time-consuming. It may lead to inaccuracies, especially for new workers. According to the Palm Oil Mill (MJM) Sdn. Bhd., the ideal Fresh Fruit Bunches (FFBs) have a yellowish and reddish outer layer, with a yellow-colored mesocarp and at least ten loose sockets [2]. These bunches should be sent to the mill within 24 hours of harvesting, but some FFBs may be downgraded or rejected due to their condition, such as unripe, empty, long stalk, dirty, poorly pollinated, damaged, small, under-ripe, and overripe bunches.

The problem with human grading is that it can lead to errors in recognizing and assessing the best condition for fresh fruit bunches of oil palm. To overcome this problem, the report proposes a system that uses smartphones to detect and evaluate the quality of fresh fruit bunches of oil palm more accurately. This will help to improve the efficiency and accuracy of oil palm fruit harvesting, which is essential for producing high-quality palm oil. The proposed system can also directly determine the quality via an android smartphone to speed up the recognition of oil palm fresh fruit bunches. As a result, the objective of this project was to create a CNN-based model capable of accurately categorizing oil palm fruit bunches. This model was utilized to create a mobile application for Android that can identify the maturity level of oil palm. Table 1 presents a comparison between oil palm research, method and dataset for neural network method use for classification from related works.

Table 1: Comparison between oil palm research, method and dataset for neural network method

Title	Method	Dataset	Advantage	Disadvantage
Palm oil classification using deep learning	CNN	Oil Palm Fruit Bunch - Unripe - Ripe	• Has an accuracy rate of 98% after 5 epochs of training • Suitable for use on image data	Lots of training data is required.
Classification of Oil Palm Fruit Ripeness Using Artificial Neural Network	ANN	Oil palm fresh fruit bunches - Ripe - Underripe - Overripe	• Has a high accuracy of 95.48% with Raman peaks • Suitable for use on tabular Data and text Data	Has a low accuracy compared to convolution neural network
Classification of oil palm fresh fruit maturity based on carotene content from Raman spectra	SVM	Oil palm fresh fruit bunches - Ripe - Underripe - Overripe	• 91.3% accuracy by 4 Peak intensity	The amount of β-carotene reduces as the fruit ripens and becomes overripe
Oil palm fruit ripeness detection using K-Nearest neighbour	KNN	Oil palm fresh fruit bunches	• Required no training time • accuracy at 65% by applying Sobel edge detection	Sorting data into categories takes longer

Thermal vision of oil palm fruits under difference ripeness quality	ANN-MLP	Ripeness Level of FFB by day - 110-130 - 131-140 - 141-170 - 171-190 - 191-200	• The thermal vision has a detection accuracy of 95.35%.	MLP has different validation accuracy.
Oil Palm Fruit Image Ripeness Classification with Computer Vision using Deep Learning and Visual Attention	CNN DenseNet	Oil palm fresh fruit bunches - Ripening - Raw - Less Ripped - Almost Ripped - Ripped - Perfectly Ripped - Too Ripped	• proved its ability in reducing vanishing gradient problem for attention module	• Resat Dense net achieved an accuracy of 69% • Too many datasets used
Maturity Grading of Oil Palm Fresh Fruit Bunches Based on a Machine Learning Approach	LDA	Oil palm fresh fruit bunches - Raw - Under-Ripe - Ripe	• Can perform with less dataset • The system achieved an accuracy of 98.88%.	misclassification regularly occurs in an under-ripe image

2. Datasets and System Overview

2.1 Datasets

Oil palm fruit bunch dataset was taken using Samsung S22 camera with 50 Megapixel with 1:1 scale ratio. The dataset was separated into three partitions which are training, testing and validation. The training data consists seventy percent of dataset that used to train the images, twenty percent of dataset used to validate the images and remain of ten percent of dataset used for training dataset to provide an accuracy of the convolution neural network model. This dataset division process is executed on Jupyter Notebook to reduce the use of compute units on google colab pro and stored automatically in a folder. On the algorithm, the input images are rescaling size pixel to 512 pixels as output image before used for learning model. The number of datasets that have been classified is recorded in the table shown in Table 2.

Table 2: Total number of datasets for training, testing and validation

Oil Palm fruit classes	Number of images	Training dataset	Testing dataset	Validation dataset
Overripe	2148	1503	216	429
Ripe	2150	1505	215	430
Underripe	2151	1505	216	430
Unripe	2149	1504	216	429

2.2 System Overview

The project consists two parts which are system development and application development as shown in Figure 1. The system development approach the Convolution Neural Network model to classify the four types of oil palm fruit bunch classes while the TensorFlow lite is used in the application development to deploy the CNN model on android smartphone to display the results.

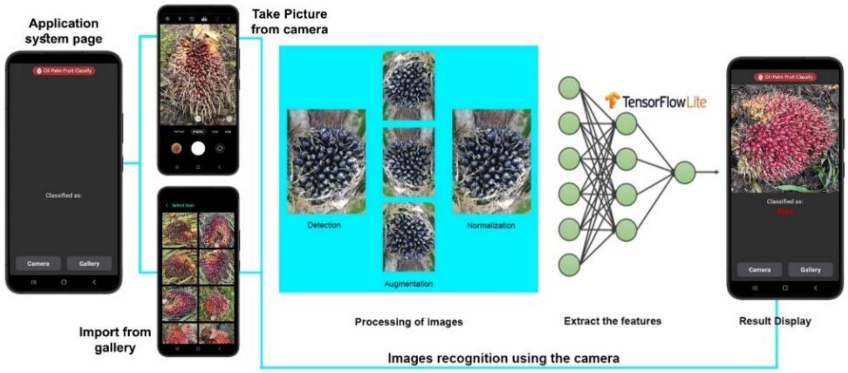

Figure 1: System configuration for classifying oil palm fruit using an android smartphone

Figure 1 shows the system overview of the oil palm fruit classification that consist of two parts which are system development and application development. The system began with a smartphone that are used to capture oil palm fruit bunch. The image data is preprocessed to develop in deep learning process. The preprocessing image includes the resizing, augmentation, data splitting, and normalization. The images are resized to 512 x 512 x 3 pixel. The data augmentation techniques such as rotate and flip, are applied to enhance the number of training technique, the preprocessed image is used to develop the deep learning by train the images with CNN model. TensorFlow Lite is used to deploy a model on an android smartphone in establishing an application development system.

2.3 Software development

Figure 2 shows the flowchart of the software development.

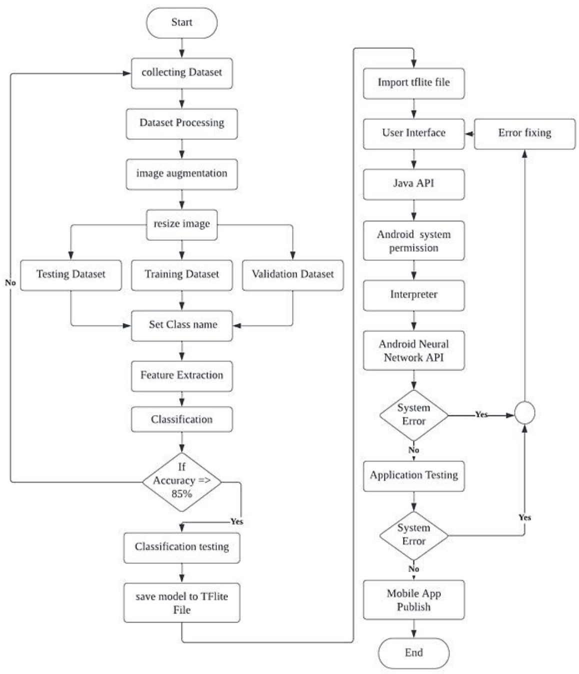

Figure 2: Flowchart of development application

2.3.2 Data preprocessing

The dataset is divided into three subsets: training, testing, and validation. These image datasets are used as the source for the dataset, and the image size is standardized across all subsets. This ensures that all images in the dataset have consistent dimensions, leading to more accurate and reliable analysis. Additionally, all images in the dataset are resized to a resolution of 512 pixels for both the width and height.

An image data generator system is employed to generate batches of tensor image data with real-time data augmentation. The training dataset for this system consists of four classes: overripe, ripe, underripe, and unripe. The TensorFlow framework includes a module called Layers, which contains classes that represent the layers of a neural network. Each layer represents a unit of computation in a neural network, and has a state (weights) that can be trained. The layers used in this neural network convolution system include 2D convolutional layer and dense fully-connected layer.

2.3.3 Convert TensorFlow Lite model

The TensorFlow Lite converter was used to optimize the trained TensorFlow model for deployment on mobile devices and building android applications. The converted TensorFlow Lite model, or TFlite model, can be downloaded on the Google Colab platform. The TensorFlow Lite converter is an efficient tool that allows for conversion of the TensorFlow model to a smaller, faster version that can be easily deployed on mobile devices.

2.3.3 Classification

These layers are stacked to form a CNN architecture. Two important parameters in CNNs are the activation function and the dropout layer. The fully-connected layer uses the results of the convolutional layer to predict the image's class based on the features extracted in earlier stages. The convolutional layer extracts and identifies the individual aspects of images for analysis in a process known as feature extraction [3]. CNNs make predictions by analyzing an image and determining whether specific features are present, and then classifying the image accordingly. However, this process requires a large amount of training data to achieve high accuracy in object detection [4].

2.4 Android application development

The Android Studio design editor is a tool that helps developers create and customize the look of their app's user interface. It makes it easy for users to use the app by providing a visual layout and tools for adding and editing elements like buttons and images. The app's main screen displays the title, logo and an image of the fruit bunch and its classification. Users can also take a picture or choose one from their gallery. The background color can be changed to light or dark depending on the user's preference.

Figure 3 shows the design of the android application. TensorFlow Lite is a tool used to add machine learning to android projects. The trained model is saved and converted into a TensorFlow Lite format file (.tflite) from Google Colab. Then it is loaded into Android Studio and executed using the Java interpreter. The interpreter works with operation kernels. For faster performance, the interpreter uses the Android Neural Networks API. This API can be used for image classification, prediction and selection.

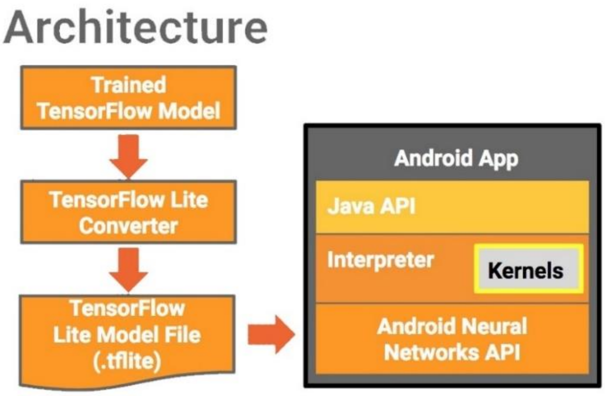

Figure 3: The TensorFlow lite architecture on android application development

3. Results and Discussion

The results and discussion section provides a detailed examination of the data and analysis conducted during the study. This section is divided into several sub-sections, including the analysis of the model's learning curve, confusion matrix, and performance evaluation. Additionally, the results of the application, detection rate, and the level of confidence in the images are also discussed in this section.

The performance of the CNN-based oil palm fruit grading application was evaluated on two different smartphones, namely the high-end Samsung S22 and the mid-range Samsung A30, to investigate the performance of smartphone specifications on the applications. By conducting this experiment, the varying specifications of smartphones can affect the performance of the application, which can be critical information for developers to consider when designing and optimizing using oil palm fruit classification applications. Therefore, the CNN-based oil palm fruit grading application's performance on both the Samsung S22 and Samsung A30 devices.

3.1 Learning graph of the convolution neural network model

The convolutional neural network system achieved high training and validation accuracy, with a maximum value of 1.0. Figure 4 illustrates the system's accuracy progress over 20 epochs, with a final training accuracy of 1.0 and validation accuracy of 0.93. Figure 5 presents the training and validation loss progression for a CNN model over 20 epochs. The final training loss recorded is 0.0051 and validation loss is 0.2001. The graph illustrates that as the number of epochs increases, both accuracy and loss improve. This model, designed for oil palm fruit bunch classification on android smartphone applications, achieved an accuracy of over 85%.

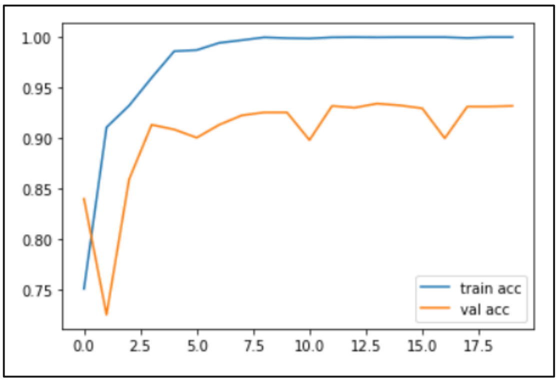

Figure 4: Graph of training and validation accuracy

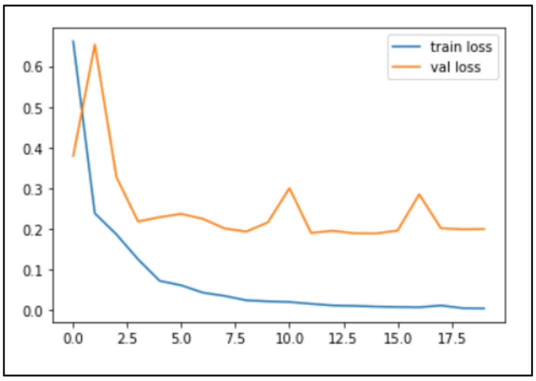

Figure 5: Graph of training and validation loss

3.2 Confusion matrix

The confusion matrix in Figure 6 has four different categories of ripeness for oil palm fruit: overripe, ripe, underripe, and unripe. The confusion matrix in Figure 5 displays the count of accurate and inaccurate predictions made by the classification model for each class. These counts can be used to calculate the accuracy of the model for each class. The matrix reveals that the model has an accuracy of 97.22% for the overripe class, 88.94% for the ripe class, 93.07% for the underripe class, and 100% for the unripe class. The overall accuracy of the model, which is the proportion of all predictions that are correct, is 93.19%. This information provides insights into the model's performance for each class and identifies any potential areas for improvement in the classification process.

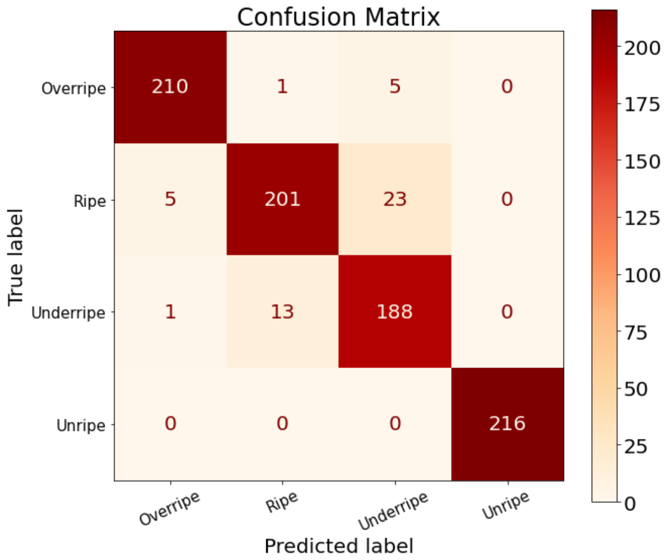

Figure 6: Confusion matrix of CNN model

3.3 Performance evaluation

Figure 7 shows the results of the classification report for the model. The classification report compares the predicted class labels to the true class labels. According to the report, the model correctly predicted 94% of the class labels. For the 'overripe' class, the model had a higher recall rate of 97% compared to 88% for the 'ripe' class and 93% for the 'underripe' class. The 'underripe' class had the highest recall rate of 100%. Additionally, the 'underripe' class had the highest F1 score of 100%, followed by 'overripe' with an F1 score of 97%, 'ripe' with an F1 score of 91%, and 'underripe' with an F1 score of 90%.

```
[ ]  #get classification report
     print(classification_report(y_pred,test_y))

               precision    recall  f1-score   support

           0       0.97      0.97      0.97       216
           1       0.93      0.88      0.91       229
           2       0.87      0.93      0.90       202
           3       1.00      1.00      1.00       216

    accuracy                           0.94       863
   macro avg       0.94      0.95      0.94       863
weighted avg       0.95      0.94      0.94       863
```

Figure 7: Classification report of model

3.4 Application result

Figure 8 presents the results of evaluating the ripeness of oil palm fruit bunches using two different Android smartphones, the Samsung Galaxy S22 (SM-S901E) and the Samsung Galaxy A30 (SM-A305F). The images were preprocessed by rescaling to a resolution of 512x512 pixels before analysis. The proposed model was used to classify the images. The figure illustrates examples of overripe, ripe, underripe, and unripe fruit classified by the model on both the SM-S901E and SM-A305F smartphones. Additionally, the inference time for both devices did not exceed 1050ms for the SM-S901E and 1140ms for the SM-A305F, with the Android application size being 81.82 MB. Figure 9 depicts the detection system for four dataset classes based on the android smartphone model.

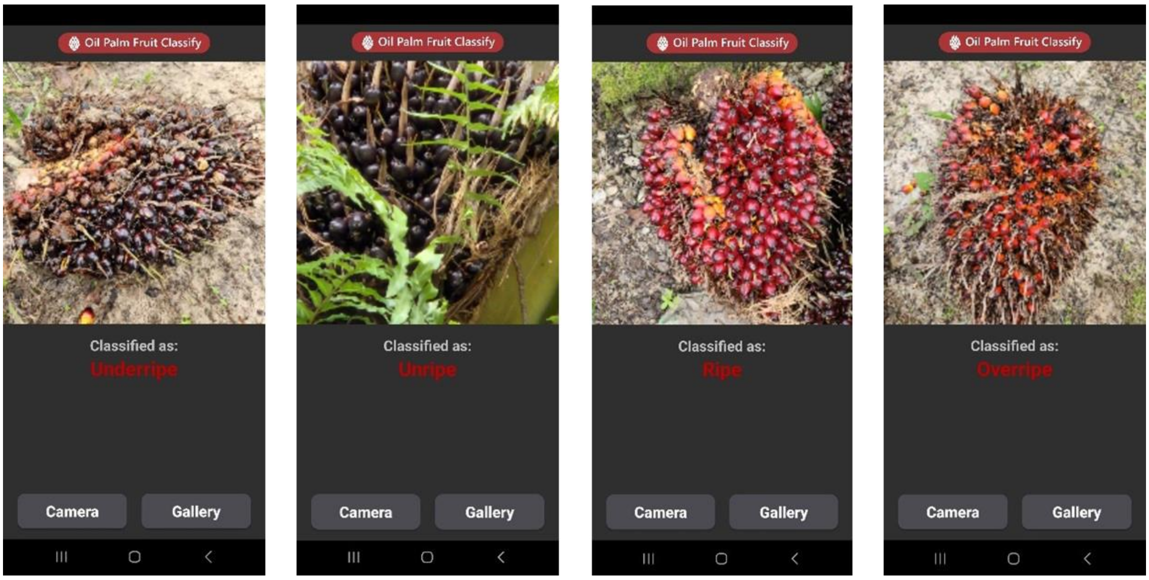

Figure 8: Result of four classes dataset using Samsung Galaxy S22 via camera

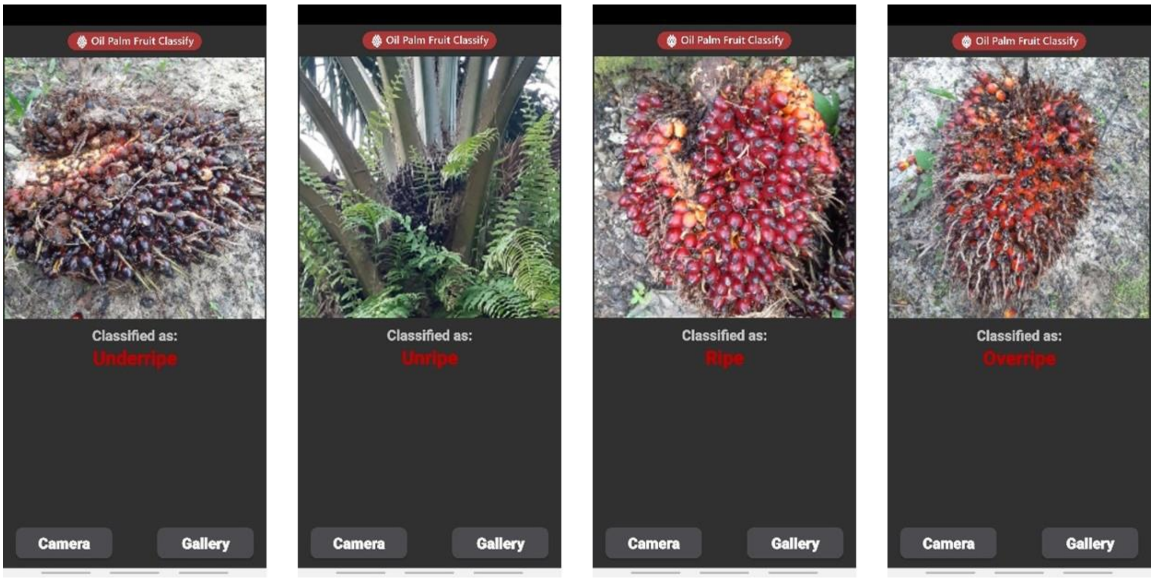

Figure 9: Result of four classes dataset using Samsung Galaxy A30 via camera

3.5 Confidence percentages

Figure 10 shows the accuracy for the overripe class got 89% for model SM-S901E, while SM-A305F got 100% for the overripe class with 19 samples of oil palm fruit bunch. The second class is the class with the most samples taken, which is 43 samples of oil palm fruit bunch. The SM-S901E model got 98%, while the SM-A305F got 86% for the ripe class. For the underripe class, a total of 18 oil palm fruit bunch samples were taken for the detection of ripeness. The coded underripe accuracy is 78% for both of these models. For unripe, accuracy is 90% for the SM-S901E and SM-A305F models get 100% accuracy for this class. The total accuracy recorded was 88.79% for the SM-S901E model and 91% for the SM-A305F model. The difference in accuracy is based on the image processing on the android smartphone model itself after taking a sample picture.

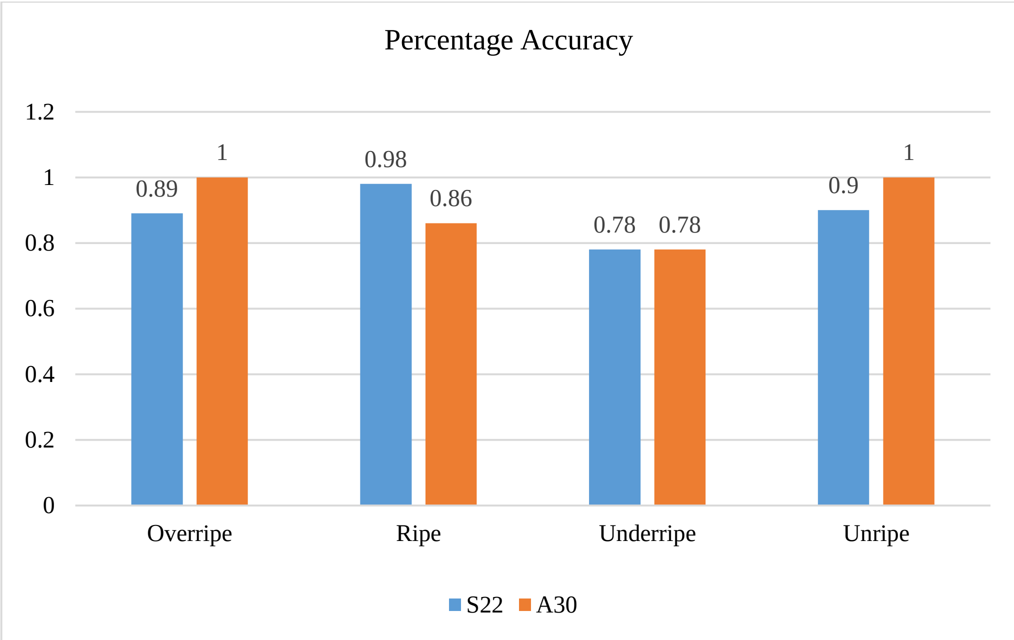

Figure 10: Percentage accuracy for 90 images using different models

The accuracy for the overripe class got 89% for model SM-S901E, while SM-A305F got 100% for the overripe class with 19 samples of oil palm fruit bunch. The second class is the class with the most samples taken, which is 43 samples of oil palm fruit bunch. The SM-S901E model got 98%, while the SM-A305F got 86% for the ripe class. For the underripe class, a total of 18 oil palm fruit bunch samples were taken for the detection of ripeness. The coded underripe accuracy is 78% for both of these models. For unripe, accuracy is 90% for the SM-S901E and SM-A305F models get 100% accuracy for this class. The total accuracy recorded was 88.79% for the SM-S901E model and 91% for the SM-A305F model. The difference in accuracy is based on the image processing on the android smartphone model itself after taking a sample picture.

4. Conclusion

The effectiveness of using a Convolutional Neural Network for this classification task by training a model on a dataset of 8458 images from the four classes. The model achieved an overall accuracy of 93.19% on the Google Colab Pro platform. An Android application successfully developed that allows users to classify the ripeness of oil palm fruit bunches using their smartphones. The results of the classification are displayed within the app's user interface. This application is compatible with a wide range of Android smartphones and has a small size of 81.82MB, making it easy to use without consuming significant storage space. The effectiveness of the smartphone application was demonstrated with a detection time of under 1200ms, depending on the performance of the device being used.

Acknowledgement

The authors would like to thank the Faculty of Electrical and Electronic Engineering for supporting the development of this project.

References

[1] FFB grading guideline. MJM (PALM OIL MILL) SDN BHD. Retrieved April 6, 2022, from http://www.mjmpom.com/ffb-grading-guideline/

[2] Malaysia prices of crude palm oil. Retrieved April 6, 2022, from https://bepi.mpob.gov.my/admin2/price_local_daily_view_cpo_msia.php?more=Y&jenis=1Y&tahun=2022

[3] Basic CNN architecture: Explaining 5 layers of Convolutional Neural Network. upGrad blog. (2021, December 9). Retrieved May 15, 2022, from https://www.upgrad.com/blog/basic-cnn-architecture/

[4] Difference between ann, CNN and RNN. GeeksforGeeks. (2020, July 17). Retrieved May 15, 2022, from https://www.geeksforgeeks.org/difference-between-ann-cnn-and-rnn/