Palm Oil Fresh Fruit Bunch Ripeness Grading Recognition Using Convolutional Neural Network

Zaidah Ibrahim[1], Nurbaity Sabri[2], Dino Isa[3]
[1]*Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA, Campus Shah Alam, 40450 Selangor*
[2]*Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA, Campus Melaka, 77300 Melaka*
[3]*Faculty of Engineering, University of Nottingham Malaysia Campus, Jalan Broga, 43500 Semenyih, Selangor*
zaidah@tmsk.uitm.edu.my

***Abstract*—This research investigates the application of Convolutional Neural Network (CNN) for palm oil Fresh Fruit Bunch (FFB) ripeness grading recognition. CNN has become the state-of-the-art technique in computer vision especially in object recognition where the recognition accuracy is very impressive. Even though there is no need for feature extraction in CNN, it requires a large amount of training data. To overcome this limitation, utilising the pre-trained CNN model with transfer learning provides the solution. Thus, this research compares CNN, pre-trained CNN model and hand-crafted feature and classifier approach for palm oil Fresh Fruit Bunch (FFB) ripeness grading recognition. The hand-crafted features are colour moments feature, Fast Retina Keypoint (FREAK) binary feature, and Histogram of Oriented Gradient (HOG) texture feature with Support Vector Machine (SVM) classifier. Images of palm oil FFB with four different levels of ripeness have been acquired, and the results indicate that with a small number of sample data, pre-trained CNN model, AlexNet, outperforms CNN and the hand-crafted feature and classifier approach.**

***Index Terms*—AlexNet; Convolutional Neural Network; Machine Learning; Palm Oil FFB Ripeness Classification.**

I. Introduction

Recognizing the correct level of ripeness of palm oil Fresh Fruit Bunch (FFB) is crucial because only the ripe palm oil FFB produces the optimum quantity of oil palm. This different ripeness can be identified by colour where red represents ripe; reddish-orange indicates overripe, reddish black is under-ripe while purplish black corresponds to unripe. Various colour models such as Red, Green, and Blue (RGB) [1] and Hue, Saturation and Intensity (HSI) [2] have been utilised by researchers to classify these various ripeness classifications. Besides that, texture feature such as Basic Gray Level Aura Matrix (BGLAM) [3] has also been applied but none of these approaches has reached 100% recognition accuracy. Furthermore, these hand-crafted features with classifier approaches require relatively extensive time in selecting the suitable feature and classifier.

In recent years, Convolutional Neural Networks (CNN) has been used in many computer vision tasks related to fruits and plants recognition such as leaves identification [4], plant species identification [5], and fruit category classification [6]. Currently, with easy access to massive data and the increase of computing power of the current hardware, outstanding results are achievable. However, there are times when it is not possible to acquire a large quantity of data. To overcome this problem, pre-trained CNN model has been developed where the training process is being performed using other images, and the learning parameters are transferred to the new classification layers for the recognition of the new small amount of data. One of the popular pre-trained CNN models is AlexNet [7] where 1.2 million high-resolution images with 1000 different classes have been used for training [8].

A comparative study of AlexNet and the handcrafted feature and classifier approach for leave identification shows that AlexNet produces better results [4]. Since there is no report regarding the application of CNN or AlexNet for palm oil FFB ripeness grading identification, this paper takes into account this issue and conducts a set of experiments to fill the gap that exists. Thus, this paper investigates the palm oil FFB ripeness grading recognition performance by comparing the results produced by the hand-crafted feature with classifier approach, CNN, and AlexNet.

This paper is organised as follows. Next section discusses the works related to CNN. Section III explains the classification methods utilised in this research. Section IV describes the presented dataset. Section V discusses the result analysis followed by a conclusion in the last section.

II. Related Work

CNN provides an excellent solution which can extract a hierarchical representation of the input data that are invariant to transformations and scales. The basic structure of a CNN is the convolutional layer, pooling layer, non-linearity layer and fully connected or classification layers. CNN was applied to various computer vision problems such as character recognition with three convolutional layers, two max-pooling layers [9], and plant disease recognition that concatenates two convolutional and average pooling layers [10].

These applications have a different architecture where they are composed of different numbers of convolutional layers and different types of pooling layers. Wu et al. [9] create a dataset of 40000 images of Chinese, Roman, and Arabic characters while [10] constructs a dataset of 1450 images of leaves from three species of apple trees. Krizhevsky et al. [8] propose AlexNet, a deep CNN that consists of five convolutional layers and three fully connected layers. It provides better classification results since deeper layers can extract more features. AlexNet has been applied by [11] to classify street view images and [12] for insect classification.

III. Classification Methods

This section discusses the three different classification methods evaluated in this paper. They are handcrafted feature and classifier, CNN and AlexNet.

A. Hand-crafted Feature and Classifier

Conventional handcrafted feature and classifier or machine learning approach requires two separate phases that are feature extraction and classification. Since colour is a significant feature to classify ripeness, it has been used for palm oil FFB ripeness. A comparative study has been conducted among various colour features and colour moment with Support Vector Machine (SVM) to produce a proper palm oil FFB ripeness classification, but the authors tested only two levels of ripeness [13]. This paper also uses a colour moment feature with SVM but to classify four levels of ripeness.

For the classification tasks where the colour feature is not significant, another feature such as texture was applied including Self-Invariant Feature Transform (SIFT), Fast Retina Keypoint (FREAK) for face recognition [17] and Histogram of Oriented Gradient (HOG) for facial expression [18]. This paper investigates the performance of FREAK as its recognition performance is comparable to SIFT and at the same time, it has low computational cost [17]. HOG has proven to achieve proper recognition results [18]. Moreover, this paper examines its performance for palm oil FFB ripeness recognition.

FREAK is a binary descriptor that simulates human vision process where higher density points are grouped at the centre of the sampling grid. It is constructed by comparing the intensity between different pairs of sampling points by thresholding differences of comparable Gaussian kernels [17]. Changing the size of the Gaussian kernels in relation with the log-polar retinal pattern and overlapping the receptive fields produce better performance. Figure 1 shows FREAK sampling pattern that is similar to the retinal cells.

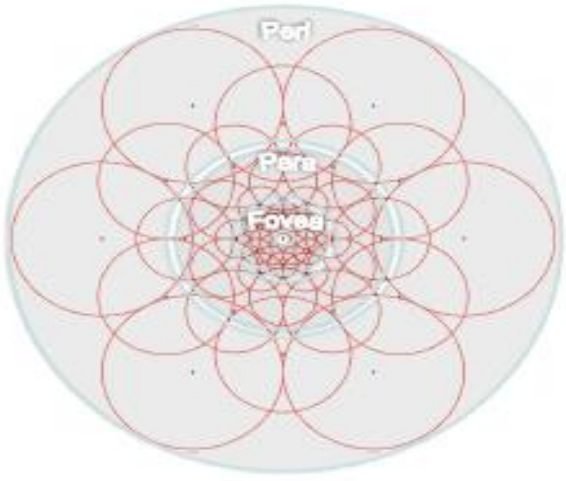


Figure 1: Illustration of FREAK sampling pattern where each circle represents a receptive field that corresponds with the Gaussian kernel [17]

HOG represents the image appearance by the distribution of local intensity gradients that are computed for each cell that is equally divided in the image [18]. The result of HOG is the normalised group of histograms that characterise the blocks or groups of adjacent cells. An illustration of the computation of HOG process is shown in Figure 2.

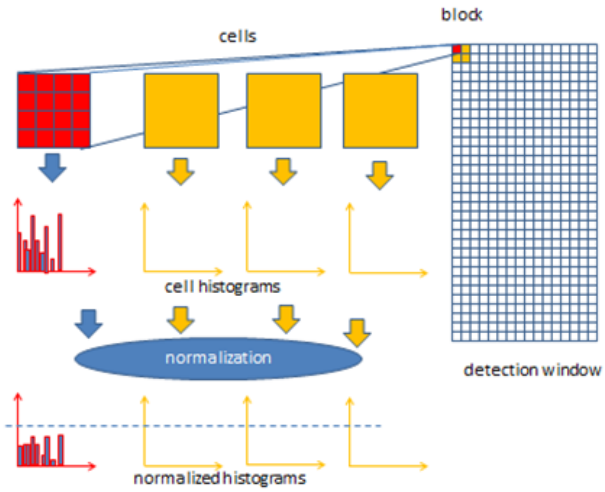


Figure 2: An illustration of the computation of HOG process [18]

One-against-all (OAA) SVM has shown a good result for FFB palm oil ripeness classification in [13], and it is being applied in this research.

B. CNN

The architecture of a typical CNN is structured as a series of layers. A stack of CNN consists of three layers that are a convolutional layer, Rectified Linear unit (ReLu) layer, pooling layer and followed by a fully connected layer (classification layer) [8]. The convolutional layer extracts features of an image by using a filter that strides over the input image and produces a feature map. The different filter produces different feature map that acts as feature detectors. Multiple convolutional layers can form different feature maps to ensure full extraction of various features.

ReLu layer replaces all negative pixel values in the feature map to zero. Pooling layer down-samples the feature map after ReLu layer to reduce the dimensionality. A typical pooling layer is max-pooling that computes the maximum of a local feature map. An average pooling layer takes the average of a local feature map. Neighbouring pooling takes input from feature maps that that are shifted or stride by more than one row or columns. This operation reduces the dimension of the feature maps and acts as invariance to distortion or small shifts. Fully connected layer performs the classification process. Figure 3 shows the CNN layers.

A pre-trained CNN model like AlexNet, also called transfer learning model, is where knowledge is learned from training a large number of datasets. It won the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) in 2012. AlexNet consists of 25 layers that combine a few stacks of convolutional layers and fully connected layers [8]. An illustration of AlexNet layers is shown in Figure 4.

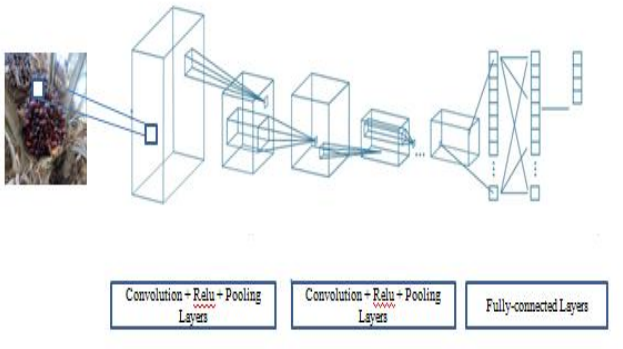


Figure 3: An illustration of CNN layers.

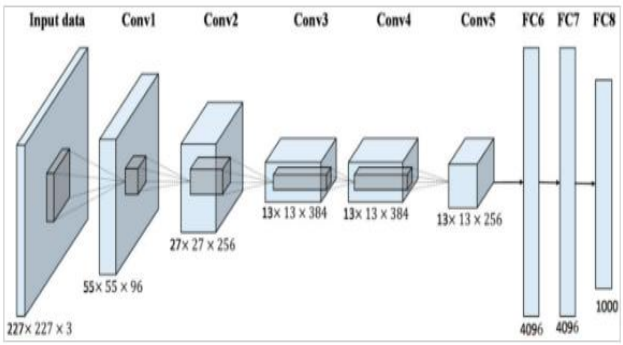


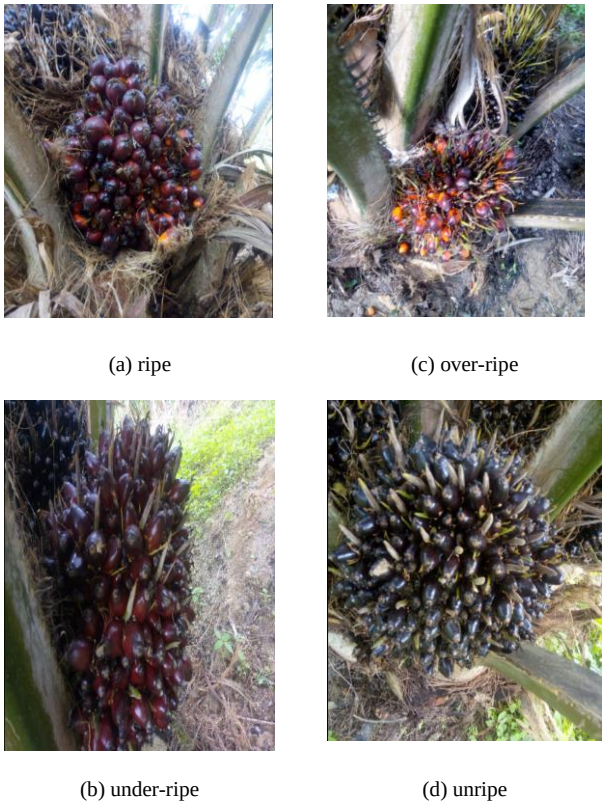
Figure 4: An illustration of the architecture of AlexNet [8].

IV. DATASET

Given the lack of publicly available datasets to support research for palm oil FFB ripeness grading recognition, we built our own image dataset. For this experiment, a total of 120 images of palm oil FFB with 30 images from each of the four levels of ripeness have been captured and labelled by a palm oil expert from Johor, Malaysia. We believe that 120 images are sufficient to produce good results for this research. They were captured during daylight, and these images are not cropped where there is no elimination of unwanted objects such as the leaves and other background images. Some sample images are shown in Figure 5.

V. RESULTS AND DISCUSSION

The experiment is conducted using Matlab 2017a. The images for the handcrafted approach were downsized to 100x100 pixels due to memory constraint. Figure 6, 7, and 8 show the coding involved in extracting HOG, FREAK and colour moment features, respectively. The colour images are converted to grayscale before the extraction of the HOG and FREAK features. The results of these features are vectors of size 900 for HOG, 64 for FREAK and 9 for a colour moment. Figure 9, 10 and 11 illustrate some sample data produced by HOG, FREAK and colour moment, respectively where each row represents the vectors of an image. The vectors from each feature are then fed into SVM for ripeness classification where 80% of the data is used for training and 20% for testing.



(a) ripe (c) over-ripe

(b) under-ripe (d) unripe

Figure 5: Sample images of palm oil FFB with four levels of ripeness (a) ripe, (b) under-ripe, (c) over-ripe, (d) unripe.

```
K=imresize(X, [100 100]);
I=rgb2gray(K);
features = extractHOGFeatures(I);
```

Figure 6: Sample coding to extract HOG features.

```
K=imresize(X, [100 100]);
I=rgb2gray(K);
corners = detectFASTFeatures(I);
features = extractFeatures(I, corners, 'Method', 'FREAK');
fr=mean(features.Features);
```

Figure 7: Sample coding to extract FREAK binary features.

```
img = imresize(im, [100 100]);
Red = double(img(:, :, 1)); Green = double(img(:, :, 2));
Blue = double(img(:, :, 3)); R_mean = mean( Red(:) );
G_mean = mean( Green(:) ); B_mean = mean( Blue(:) );
R_std  =std( Red(:) ); G_std  =std( Green(:) ); B_std = std( Blue(:) );
R_skew = skewness( Red(:) ); G_skew  =skewness( Green(:) );
B_skew  =skewness( Blue(:) );
```

Figure 8: Sample coding to extract colour moment features.

	1	2	3	4	5
1	0.0136	0.0015	0	0	0.0021
2	0.0330	0.0090	0.0038	0.0050	0.0306
3	0.0478	0.0125	0.0203	0.0194	0.0133
4	0.0212	0.0234	0.0202	0.0356	0.0250

Figure 9: Sample data produced by HOG feature

	1	2	3	4	5
1	94.6056	110.3521	133.3239	140.3803	100.8169
2	113.7245	127.3776	130.6837	129.2449	113.2653
3	112.9867	130.3333	142.3333	131.9467	116.3067
4	109.9130	135.0217	140.8478	150.3478	93

Figure 10: Sample data produced by FREAK feature

	A	B	C	D	E
1	81.3358	22.72587	80.64086	78.20303	25.80447
2	71.42692	13.94613	53.6691	78.29349	39.20291
3	111.8412	28.42008	96.9835	111.5603	35.58592

Figure 11: Sample data produced by colour moments

For experiments using CNN and AlexNet, the images were downsized to 28x28 pixels for CNN while 227x227 pixels for AlexNet since these are the sizes required in order to run CNN and AlexNet in Matlab. CNN and AlexNet take the raw colour images, and the features are automatically extracted by the layers. Figure 12 shows the coding for the execution of CNN for one stack of layers that consist of convolve layer, ReLu layer, and pooling layer while additional stacks of layers can be added to compare the performance.

The size of the filter in the convolve layer and the value of *stride* in the pooling layer that represents the number of columns to be skipped for the sliding window can be changed as these values can affect the results of the recognition performance. Besides that, the values of *maxepochs* that represents the number of iterations for the training process and *initial learning rate* that represents the value of the weight to be adjusted during the training process can also be changed to improve the performance.

Figure 13 shows some sample coding that applies AlexNet whereby the parameters in layer 23 and 25 needs to be changed based on user's data. For instance, since there are four types of ripeness in this research, the parameter in layer 23 is set to 4.

```
layers = [imageInputLayer([28 28 3])
          convolution2dLayer(9,40)
          reluLayer
          maxPooling2dLayer(3,'Stride',3)
          fullyConnectedLayer(10)
          softmaxLayer
          classificationLayer()];
options = trainingOptions('sgdm','MaxEpochs',15,
   'InitialLearnRate',0.001);
```

Figure 12: Sample coding to compute CNN

```
alex=alexnet;
layers=alex.Layers;
layers(23)=fullyConnectedLayer(4);
layers(25)=classificationLayer
allImages=imageDatastore('myImages', 'IncludeSubfolders', true,
'LabelSource', 'foldernames');
[trainingImages, testImages] = splitEachLabel(allImages, 0.8,
'randomize');
opts=trainingOptions('sgdm', 'InitialLearnRate', 0.001, 'MaxEpochs', 20,
'MiniBatchSize', 20);
myNet=trainNetwork(trainingImages, layers, opts);
```

Figure 13: Sample coding to apply AlexNet

Table 1 shows the results of CNN with one stack (1 convolutional layer, 1 Relu layer and 1 pooling layer) and two stacks of layers as listed in the first column.

Accuracy is computed by counting the number of the correctly recognised image in the testing data.

Total time is the amount of time measured in seconds to compute the whole process. By looking at the last two columns in Table 1, we can see that CNN with one stack of the layer can still achieve similar results with two stacks of layers, which is 0.92 for this experiment but the processing time is increased as the number of layers increases. It is also observed that a smaller learning rate reduces processing time.

Since AlexNet is a pre-trained CNN model, not many parameters can be fine-tuned compared to CNN. Table 2 illustrates the results of AlexNet where the accuracy of 1 is achieved. However, since AlexNet has more layers compared to CNN that is applied in this research, the processing time is also higher compared to CNN as listed in Table 1.

By referring to Table 1 and Table 2, it is observed that an increase in the accuracy is related to the increase in the depths of the network but at the same time increases the processing time. The results displayed by all the three tables indicate that by using AlexNet, an excellent result can be achieved even though the number of our training data is small. This is due to the tremendous number of training data that AlexNet has used during training, and the learning parameters can be transferred for other recognition purposes that has a small amount of data. The user is released with the burden to experiment with various features and classifiers to achieve a good result. With AlextNet, the user only needs to input the colour images, and AlexNet will automatically extract the features and perform the recognition.

Table 3 lists the results produced by the handcrafted features with SVM classifier where the best accuracy result that can be obtained is only 0.75 with HOG. Since HOG generates a vector of size 900 for each image, the processing time is higher compared to FREAK and colour moment. Even though the colour is a useful feature that can describe the ripeness of palm oil FFB, the results produced by colour moment is not as good as the other features since colour can be easily influenced by illumination. The result can be improved if the image is cropped to obtain the image of the palm oil FFB only without as shown in Figure 14, but this manual cropping is time-consuming while automatic segmentation and cropping may not be accurate.

Figure 14: Sample images cropped images of palm oil FFB

Table 1
The Performance of CNN for Palm Oil FFB Grading Ripeness Recognition

Size of Conv Layers	Stride	Learning Rate	Total Time (s)	Accuracy (%)
9x40	3	0.001	2.376	0.77
5x20	3	0.001	1.857	0.92
5x20	3	0.0001	1.858	0.87
9x40 and 5x20	3	0.001	2.279	0.72
9x40 and 5x20	2	0.001	2.32	0.82
9x60 and 5x40	2	0.001	3.402	0.92

Table 2
The Performance of AlexNet for Palm Oil FFB Grading Ripeness Recognition

Learning Rate	Total Time (s)	Accuracy (%)
0.001	84.47	1
0.0001	50.61	1

Table 3
The Performance of Handcrafted Features with SVM for Palm Oil FFB Grading Ripeness Recognition

Features	Total Time (s)	Accuracy (%)
HOG	78.59	0.75
FREAK	76.94	0.71
Color Moment	70.05	0.67

VI. CONCLUSION

We have presented a comparative study between handcrafted feature and classifier approach that consists of three different features namely colour moment, FREAK and HOG with SVM classifier, CNN and pre-trained CNN that is AlexNet, concerning accuracy and processing time. The performance of the CNN depends on the number of training data and the number of layers. Applying CNN from scratch require a tremendous amount of training data to achieve relatively good results. A deep layer can lead to better results but at a slow processing time. The experimental results indicate that AlexNet outperforms the other two approaches since it has more layers where more features can be extracted but with higher processing time. The use of AlexNet is suitable for classification tasks where a large amount of data is not available and for tasks where high processing time is not an issue. Future works include experimentations with other deeper pre-trained CNN models that are GoogleNet, VGG-16, ResNet, and Inception.

ACKNOWLEDGEMENT

The authors gratefully acknowledge Universiti Teknologi MARA for sponsoring this research under Lestari grant 600-IRMI/MyRA 5/3/LESTARI (060/2017).

REFERENCES

[1] Roseleena, J., Nursuriati, J., Ahmed, J., and Low, C. Y., "Assessment of palm oil fresh fruit bunches using a photogrammetric grading system," *International Food Research Journal* 18(3), pp. 999-1005, 2011.

[2] M. K. Shabdin, A.R. Mohamed Shariff, M. N. Azlan Johari, N. K Saat, and Z. Abbas, "A study on the oil palm fresh fruit bunch (FFB) ripeness detection by using Hue, Saturation and Intensity (HIS) approach", 8th IGRSM *International Conference, and Exhibition on Remote Sensing& GIS*, 2016.

[3] M. S. M. Alfatni, R. Shariff, M. Z. Abdullah, and O. M. B. Saaed, "Recognition System of Oil Palm Fruit Bunch Types Based on Texture and Image Processing," *Journal of Computational and Theoretical Nanoscience* 19(12), pp. 3441-3444, 2013.

[4] M. A. Hedhazi, I. Kourbane, and Y. Genc, "On Identifying leaves: A Comparison of CNN with classical ML methods," IEEE *25th Signal Processing and Communications Applications Conference*, 2017.

[5] S. H. Lee, C. S. Chan, P. Wilkin, and P. Remagnino, "Deep-plant: Plant identification with convolutional neural networks," *IEEE International Conference on Image Processing (ICIP)*, 2015.

[6] Y. D. Zhang, Z. D. Dong, X. Chen, W. Jia, S. Du, K. Muhammad and S. H. Wang, "Image-based fruit category classification by 13-layer deep convolutional neural network and data augmentation," *Journal of Multimedia Tools and Applications*, Springer, pp. 1-20, 2017.

[7] Y. LeCun, Y. Bengio, and G. Hinton, "Deep Learning,". Nature 521, pp. 436-444, 2015.

[8] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Networks." *Advances in Neural Information Processing Systems* 25, 2012.

[9] P. Wu, Z. Huang, and D. Li, "Research on the Character Recognition for Chinese License Plate Based on CNN," 3rd IEEE *International Conference on Computer and Communication*, 2017.

[10] L. G. Nachtigall, R. M. Araujo, and G. R. Nachtigall, "Classification of Apple Tree Disorders Using Convolutional Neural Networks", IEEE *International Conference on Tools with Artificial Intelligence*," 2016.

[11] Q. Wang, C. Zhou, and N. Xu, "Street View Image Classification based on Convolutional Neural Network", 2nd *IEEE Advanced Information Technology, Electronic and Automation Control Conference*, 2017.

[12] S. Lim, S. Kim, and D. Kim, "Performance Effect Analysis for Insect Classification using Convolutional Neural Network," *7th IEEE International Conference on Control System, Computing and engineering*, 2017.

[13] N. Sabri, Z. Ibrahim, S. Syahlan, N. Jamil, and N. A. Mangshor, "Palm Oil Fresh Fruit Bunch Ripeness Grading Identification using Color Features," *Journal of Fundamental and Applied Sciences*, 2017.

[14] J. Krizaj, V. Struc, S. Dobrisek, D. Marcetic, and S. Ribaric, "SIFT vs FREAK: Assessing the usefulness of two Keypoint descriptors for 3D face verification," *37th International Convention on Information and Communication Technology*, Electronics and Microelectronics, 2014.

[15] S. An, and Q. Ruan, "3D facial expression recognition algorithm using local threshold binary pattern and histogram of oriented gradient," IEEE *13th International Conference on Signal Processing*, pp. 265-270, 2016.

[16] A. Mary, M. O. Chacko, and P. M. Dhanya, "A Comparative Study of Different Feature Extraction Techniques for Offline Malayalam Character Recognition," *Computational Intelligence in Data Mining*, vol. 2, Springer, 2015.

[17] A. Alahi, R. Ortiz, and P. Vandergheynst, "FREAK: Fast Retina Keypoint," *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[18] N. Dalal, and B. Triggs, "Histograms of Oriented Gradients for Human Detection," *IEEE Conference on Computer Vision and Pattern Recognition*, 2005.