Banana Ripeness Classification Based on Deep Learning using Convolutional Neural Network

Conference Paper · April 2021

DOI: 10.1109/EIConCIT50028.2021.9431928

2 authors:

Raymond Erz Saragih
Universal University

Andi Wahju Rahardjo Emanuel
Universitas Atma Jaya Yogyakarta

Banana Ripeness Classification Based on Deep Learning using Convolutional Neural Network

Raymond Erz Saragih
Magister Informatika
Universitas Atma Jaya Yogyakarta
Yogyakarta, Indonesia
195303110@students.uajy.ac.id

Andi W. R. Emanuel
Magister Informatika
Universitas Atma Jaya Yogyakarta
Yogyakarta, Indonesia
andi.emanuel@uajy.ac.id

***Abstract*—Fruit ripeness is an important thing in agriculture because it determines the fruit's quality. Determining the ripeness of the fruit that was done manually poses several weaknesses, such as takes a relatively long time, requires a lot of labor, and can cause inconsistencies. The agricultural sector is one of the essential sectors of the economy in Indonesia. However, sometimes the process of determining fruit ripeness is still done by using the manual method. The development of computer vision and machine learning technologies can be used to classify fruit ripeness automatically. This study applies the Convolutional Neural Network to classify the ripeness of the banana. The banana's ripeness is divided into four classes: unripe/green, yellowish-green, mid-ripen, and overripe. Two pre-trained models are used, which are MobileNet V2 and NASNetMobile. The experiment was conducted using Google Colab and several libraries such as OpenCV, Tensorflow, and scikit-learn. The result shows that MobileNet V2 achieves higher accuracy and faster execution time than the NASNetMobile. The highest accuracy achieved is 96.18%.**

Keywords— Fruit ripeness, computer vision, CNN, pre-trained model

I. INTRODUCTION

Advances in technology bring a significant impact on human life. Technology is used in various fields to assist humans in carrying out different processes. One of the implications of technology is in agriculture. In agriculture, the use of innovative technology is one of the crucial aspects [1]. Some of the latest technologies used are computer vision and machine learning [1], [2]. Computer vision and machine learning in agriculture are used for several tasks, such as fruit detection [3], classifying fruits [4], and determining the level of ripeness of the fruit or determining fruit defect [5].

Fruit ripeness is an essential thing in agriculture because it determines the fruit's quality [5]. Previously, determining the ripeness of the fruit was done manually. There are several weaknesses in the manual method. It takes a relatively long time to do it, requires a lot of labor, and can cause inconsistencies in determining fruit ripeness [5], [6], [7]. The emergence of computer vision technology can overcome these problems because the classification of fruit ripeness can be done automatically; therefore, it is relatively quick, consistent, and relatively inexpensive [2].

In Indonesia, agriculture plays a vital role in the country's economy [8], [9]. In Indonesia, various kinds of fruit are produced, such as mango, Salak, orange, banana, watermelon, and many more [10]. However, sometimes the process of determining fruit ripeness is still done by using the manual method. The development of computer vision and machine learning technologies can be used to classify fruit ripeness automatically. Within machine learning, deep learning technology has developed, and one of the deep learning methods that play a significant role is the Convolutional Neural Network (CNN).

This study aims to apply the Convolutional Neural Network to classify the ripeness of the banana. The levels of ripeness measured are unripe/green, yellowish-green, mid-ripen, and overripe. Image preprocessing such as removing noise in the image and resizing were used, and data augmentation is used to produce variations of the images in the dataset. The dataset used in this study is a dataset of banana ripeness levels provided by [11]; by using the pre-trained CNN models used in this study are the MobileNet V2 [12] and NASNetMobile [13]. Transfer learning by fine-tuning is used to the pre-trained MobileNet V2 and the NASNetMobile for classifying the banana ripeness. The transfer learning method is chosen because it requires an immense computational and memory resource and many datasets to train a CNN model from scratch [14]. After training, several measurements are conducted to determine the model's performance, such as the accuracy, precision, recall, and F1 score.

II. LITERATURE REVIEW

Various studies have been conducted to determine the level of fruit ripeness using computer vision and machine learning. Kangune *et al.* [15] compare the accuracy of using CNN and Support Vector Machine (SVM) to classify grapes' maturity. Color features such as RGB and HSV and morphological features were used, and Gaussian blur was used in the preprocessing stage. The accuracy using CNN was 79.49%, higher than the SVM, which was 69%. Ibrahim *et al.* [16] used AlexNet and SVM to classify the oil palm fruit bunch's ripeness. SVM's input features are color moment, Fast Retina Keypoint (FREAK) binary feature, and Histogram of Oriented Gradient (HOG) texture feature. Preprocessing was performed, such as resizing the images and converting them to grayscale. The results obtained are that the accuracy using AlexNet is higher than using SVM. Momeny *et al.* [17] classify cherries based on their shape. The accuracy of CNN was compared with various machine learning methods, such as K-Nearest Neighbors, Artificial Neural Network (ANN), Fuzzy, and Ensemble Decision Trees. Feature extractors, such as HOG and LBP, were used. In the preprocessing stage, segmentation and resizing of the images were conducted, and data augmentation was used to increase the number of images used in the dataset. The accuracy of CNN reaches up to 99%, which is higher than other methods.

The study by Mazen and Nashat classified banana ripeness using several machine learning methods [11]. The methods include ANN, SVM, Naïve Bayes, KNN, Decision Tree, and Discriminant Analysis. Several preprocessing steps were carried out, such as image smoothing, color channel conversion, morphological filters, and segmentation. Tamura texture features and new features were used to define the banana ripeness factor and as the input. The accuracy achieved is up to 100% for the green and overripe bananas and 97.75% for yellowish-green and mid-ripen bananas. Thakur *et al.* tried to classify strawberries from their ripeness [18]. The CNN model was used to classify these strawberries' ripeness and achieve an accuracy rate of 91.6%. Pardede et al. [10] researched to classify the ripeness of various fruits, such as apple, mango, orange, and tomato. For example, color features, such as RGB, HSV, HSL, and L * a * b * were used, and SVM as the classifier. Different degree of the polynomial kernel was used to obtain the best result. The highest accuracy was obtained using the HSV color feature and 6th-degree polynomial, which is 76%.

Khojastehnazhand *et al.* used LDA and QDA to classify apricots' ripeness and estimate the fruit's volume [2]. The color features, such as G channel, grayscale, L *, and b *, were used as input for the LDA and QDA. In the preprocessing stage, noise removal and segmentation were conducted. Results show that the highest accuracy was achieved by using QDA. Suban *et al.* classify the ripeness of Papaya Carica fruit using KNN [9]. The RGB values were used as the input feature for KNN. The level of accuracy that was successfully obtained using the KNN was 100%. Arakeri and Lakshmana conducted research using ANN to classify tomatoes based on their ripeness and degree of the defect. The accuracy rate obtained was 100% for the defect and 96.47% for ripeness [6]. Castro *et al.* [5] tried to classify Cape Gooseberry fruit based on maturity level, using classifiers, such as SVM, ANN, Decision Tree, and KNN, and color features RGB, HSV, L*a*b*, were used. Besides, PCA was used to combine the color features. Results show that the L*a*b* color features and SVM achieved the highest f-measure, and the use of PCA proved to improve the model's performance.

This study aims to apply and evaluate two pre-trained CNN models: MobileNet V2 and NASNetMobile, classifying different ripeness of the banana fruit, such as unripe/green, yellowish-green, mid-ripen, and overripe. Image processing was conducted to reduce noise, and data augmentation was applied to add variations to the dataset.

III. Research Methodology

Fig. 1 shows the stages in this study: literature study, obtaining the dataset, image preprocessing, train CNN models, and evaluation.

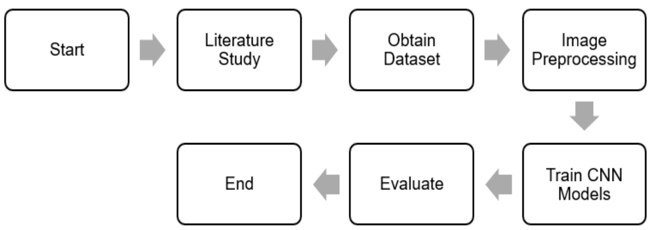

Fig. 1. Research method

Based on Fig. 1, the first stage is the literature study. In this stage, research documents and reports related to the topic are collected from various journals. The research documents and information are then used as references for this study. Through the literature study, different classifying fruit ripeness methods can be compared and become the basis for selecting the method used in this study. CNN was chosen because previous studies stated that it could achieve high accuracy.

The second stage is obtaining the dataset. The dataset used in this study is images combined from [11] and [19]. The dataset consists of 436 banana images with four ripeness classes: unripe/green, yellowish-green, mid-ripen, and overripe. The next stage is processing the images from the dataset. Image processing was done to remove noise. Two pre-trained models are used in the training stage: MobileNet V2 [12] and NASNetMobile [13]. In the Keras available models, MobileNet V2 has the smallest size among other models with the size of 14 MB, while the MobileNet has the size of 16 MB and the NASNetMobile has the size of 23 MB. Despite having a small model size, MobileNet V2 and NASNetMobile could achieve high image recognition accuracy [12], [13]. Therefore, MobileNet V2 and NASNetMobile were used in this study to create an efficient model with good performance in classifying banana ripeness. Following the training, the next step is to evaluate the models. The evaluation was done by measuring the performance of each model. Accuracy, precision, recall, specificity, sensitivity, and F-measure were used to measure the performance, and the result could determine which model performs better.

A. Dataset Description

The dataset used in this study consisted of 436 images of banana fruit and was obtained from [11] and [19]. The images were divided into four ripeness classes: unripe/green, yellowish-green, mid-ripen, and overripe. The image size varies from 225 x 225 pixels to 960 x 540 pixels. Fig. 2 shows several images of the banana dataset.

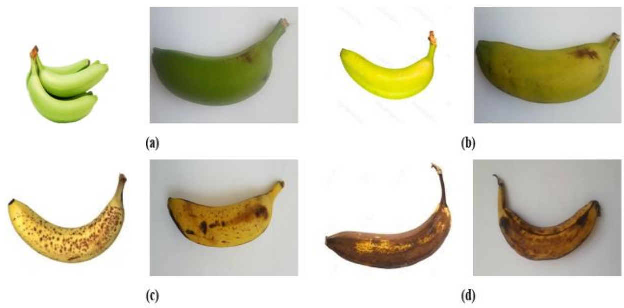

Fig. 2. Images of banana with different ripeness, (a) unripe/green, (b) yellowish-green, (c) mid-ripen, (d) overripe

B. Image Preprocessing

In this stage, a filter is applied to the images to remove noise. The filter used in this study is the bilateral filter. The process of applying the bilateral filter is shown in Fig. 3.

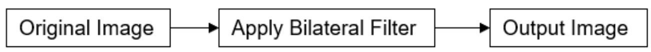

Fig. 3. Process of applying the bilateral filter

The original images from the dataset as well as test images are processed by applying the bilateral filter. The bilateral filter in this work utilizes the function provided by OpenCV. The bilateral filter can remove noise from an image while still preserving the edges of the object in the image; therefore, the object's shape is preserved [20]. The bilateral filter will produce a new output image that will then be used as the CNN model's input. An example of applying the bilateral filter to an image is shown in Fig. 4, using the images [11].

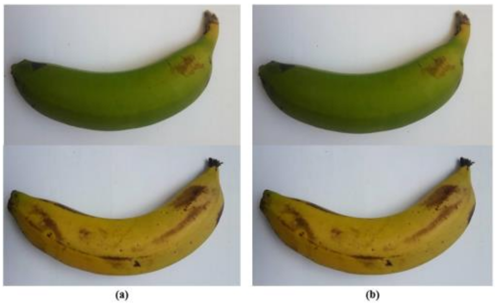

Fig. 4. Result of the bilateral filter (b) from the original image (a)

Applying the bilateral filter to an image creates a new image that has been smoothened, and the noise has been reduced; however, the shape of the banana is still preserved. As stated above, the dataset consists of images with different sizes; therefore, after applying the bilateral filter, the resulting images are resized to the size of 224 x 224 pixels following the default input image of MobileNet V2 and NASNetMobile.

C. Training CNN

After the images were processed, the next stage is training the CNN model. The CNN models used are the pre-trained MobileNet V2 [12] and NASNetMobile [13]. In this work, the MobileNet V2 and NASNetMobile used are provided by the Keras library. MobileNet V2 currently is the smallest model provided in the Keras application library, while the NASNetMobile is the third smallest model. Both models were trained on ImageNet to classify 1000 classes of objects. The original MobileNet V2 consists of 156 layers with 3,538,984 total parameters. On the other hand, the original NASNetMobile has 771 layers with 5,326,716 total parameters. As can be seen, the NASNetMobile has more in-depth architecture and more parameters than the MobileNet V2. However, in this work, the original classification layer based on the number of classes in the ImageNet dataset will not be used., thus reducing the layers of the MobileNet V2 into 154 layers and the NASNetMobile into 769 layers.

Transfer learning was applied in this study to train both models in classifying banana ripeness. Transfer learning consists of two approaches. The first approach uses a pre-trained CNN as a feature extractor for the new classification task by removing the last fully connected layer. A classifier is then trained on the features extracted from the CNN [21]. The second approach is by fine-tuning a pre-trained CNN, training a new classification layer, and retraining several layers of the pre-trained CNN model [22]. In this study, the pre-trained MobileNet V2 and NASNetMobile are fine-tuned to classify the banana ripeness. The overview of the transfer learning used in this study is shown in Fig. 5.

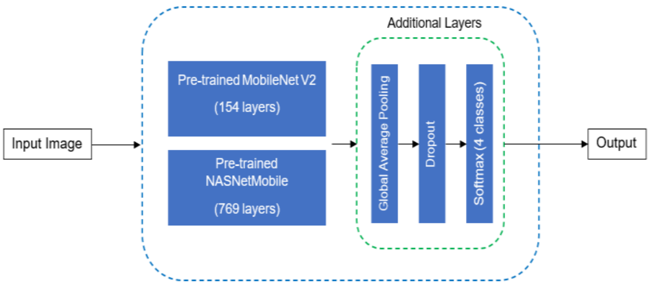

Fig. 5. Overview of the transfer learning

As shown in Fig. 5 above, the top layers of the original pre-trained MobileNet V2 and NASNetMobile were removed. A global average pooling layer was added on top of the base model of MobileNet V2, and the NASNetMobile and the final layer is the prediction layer, which uses the Softmax activation function [14]. The final classification layer is customized according to the desired number of classes, in this case, are four classes, which are green (unripe banana), yellowish-green, mid-ripen, and overripe bananas. A dropout layer with a probability of 0.3 is added between the global average pooling layer and the Softmax layer to reduce overfitting [23].

D. Evaluation of Model

The trained models then must be evaluated to know the performance. The performance is measured in terms of accuracy, precision, recall, and F1 score. The accuracy, precision, recall, and F1 score are calculated using the following equation:

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN} \tag{1}$$

$$Precision = \frac{TP}{TP+FP} \tag{2}$$

$$Recall = \frac{TP}{TP+FN} \tag{3}$$

$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision+Recall} \tag{4}$$

The scikit-learn library provides a method to calculate accuracy, precision, recall, and F1 score automatically. Therefore, in the experiment, the scikit-learn is used to give the performance result of each model.

IV. Experimental Result

The experiments were conducted using the dataset from [11] and [19]. The combined dataset contains 436 images of bananas, with four ripeness classes: green, yellowish-green, mid-ripen, and overripe. The experiments were done using Google Colab with GPU hardware accelerator and several libraries, such as OpenCV, Tensorflow, and scikit-learn. The dataset is split into training data and testing data. The training data is 70% of the dataset, and the testing data is 30% of the dataset. The splitting was done randomly. Each model was trained for 50 and 100 epochs with a batch size equal to 10. The optimizer used for each model is Adam, and the loss function used is categorical cross-entropy. Data augmentation such as horizontal and vertical flip, brightness, zoom, shear, rotation, and shifting were applied for the training data to create variations of the training images.

The first step is to freeze the base model and do a warm-up train for the new classification layer [23]. The initial epochs used for the warm-up stage are 20 epochs. The next step is to unfreeze several top layers from the base model and retrain the model with the new classification layers [14]. The unfreezing of MobileNet V2 was done from the 100th and 125th layers of the base model, while for the NASNetMobile, the unfreezing was done from the 600th and 700th layers of the base model. The learning rate used in the warm-up stage is 0.0001, and in the fine-tuning stage, the learning rate is lowered to 0.00001.

Following the training process, an evaluation was done to know the performance of each model. Table 1 shows the performance result of the MobileNet V2 model after the training process. The precision, recall, and F1score listed are based on the average macro result.

TABLE I. PERFORMANCE RESULT OF MOBILENET V2

No. of Epochs	Unfreezing Starting Layer	Accuracy	Precision	Recall	F1 score
50	100^{th}	94.66%	95.25%	94.53%	94.41%
	125^{th}	91.60%	91.80%	91.59%	91.37%
100	100^{th}	96.18%	96.53%	96.09%	96.02%
	125^{th}	93.13%	93.21%	93.08%	93.22%

As shown in Table 1, with a total of 50 epochs and unfreezing from the 100^{th} layer, the model achieves 94.66% accuracy, 95.25% precision, 94.53% recall, and 94.41% F1 score. However, by increasing the starting layer for unfreezing to the 125th layer, the accuracy, precision, recall, and F1 score are lower than the previous, 91.60%, 91.80%, and 91.59%, and 91.37%, respectively. The result shows that increasing the starting layer for unfreezing the model lowers the model's performance in classifying the banana's ripeness.

The next experiment is using a total of 100 epochs to train the model. The result shows that the model's performance is increasing than the previous experiment, with 96.18% accuracy, 96.53%% precision, 96.09% recall, and 96.02% F1 score. Like the previous experiment, by increasing the starting layer for unfreezing to the 125^{th} layer, the model's performance lower than when choosing the 100^{th} layer for the starting layer. Using 100 epochs with the 125^{th} layer as the starting layer, however, is higher than using 50 epochs with the same starting layer: 93.13% accuracy, 93.21% precision, and 93.08% recall, and 93.22% F1 score.

In the experiment, choosing the 100^{th} layer as the starting layer for unfreezing the model gives better performance on both 50 epochs and 100 epochs, with the highest achieved using 100 epochs. However, increasing the starting layer from the 100^{th} to the 125^{th} tends to reduce the model's performance.

The next result shows the performance of NASNetMobile, which can be seen in Table 2 below. The experiment was done using similar epochs to the previous experiment on MobileNet V2, which are 50 epochs and 100 epochs. For the NASNetMobile, the starting layer for unfreezing the model is the 600^{th} and 700^{th} layer because the NASNetMobile has more layers than the MobileNet V2.

TABLE II. PERFORMANCE RESULT OF NASNETMOBILE

No. of Epochs	Unfreezing Starting Layer	Accuracy	Precision	Recall	F1 score
50	600^{th}	90.08%	90.20%	90.35%	90.15%
	700^{th}	88.55%	88.75%	88.76%	88.57%
100	600^{th}	90.84%	90.81%	91.02%	90.84%
	700^{th}	90.08%	90.22%	90.35%	90.11%

Training the NASNetMobile using 50 epochs and choosing the 600^{th} layer as the starting layer for unfreezing achieves 90.08%, 90.20% precision, 90.35% recall, and 90.15% F1 score. Increasing the starting layer to the 700^{th} layer, on the contrary, achieves lower performance than the previous, which are 88.55%, 88.75%, 88.76%, 88.57% on the accuracy, precision, recall, and F1 score, respectively. The resulting pattern is like the previous result of using the MobileNet V2; increasing the starting layer for unfreezing tends to lower its performance.

The following experiment is by increasing the epochs in training to 100 epochs. Using the 600^{th} layer as the starting layer for unfreezing, the model achieves slightly higher performance than training using 50 epochs using the 600^{th} and 700th layers. The accuracy, precision, recall, and F1 score are 90.84%, 90.81%, 91.02%, and 90.84%. The next experiment uses 100 epochs and increases the starting layer for unfreezing to the 700^{th} layer. The model achieves higher performance than by using 50 epochs. The accuracy reaches by the model is 90.08%, with 90.22% precision, 90.35% recall, and 90.11% F1 score. This result is nearly like the experiment using 50 epochs and the 600^{th} layer, with only a difference in precision and F1 score.

Overall, in the experiment of using NASNetMobile, the highest performance can be achieved by using 100 epochs and choosing the 600^{th} layer as the starting layer for unfreezing the model. Like the previous experiment using the MobileNet V2, increasing the starting layer for unfreezing the NASNetMobile, tends to lower the performance in classifying the banana's ripeness.

Comparing between the MobileNet V2 and NASNetMobile, the MobileNet V2 achieves higher performance than the NASNetMobile. However, both models achieve the highest performance when training the model using 100 epochs; thus, increasing the number of epochs in training will result in better performance. Several trials are conducted using images never seen by the model to know whether the model can predict a banana's ripeness in an image and the execution time of both models when using CPU or GPU. Fig. 6 shows the test images, while the results are shown in Table 3 for MobileNet V2 and Table 4 for NASNetMobile.

(a) (b) (c) (d)

Fig. 6. Test images of banana with different ripeness, (a) unripe/green, (b) yellowish-green, (c) mid-ripen, and (d) overripe

TABLE III. PREDICTION RESULT AND EXECUTION TIME OF MOBILENET V2

Test Images	Prediction	Execution Time	
		CPU	GPU
Unripe/Green	Unripe/Green	0.093 s	0.046 s
Yellowish Green	Yellowish Green	0.098 s	0.043 s
Mid-Ripen	Mid-Ripen	0.091 s	0.042 s
Overripe	Overripe	0.125 s	0.081 s

The results in Table 3 show that the MobileNet V2 can predict all the banana ripeness correctly. When the prediction is executed using CPU, the execution time is in the range of 0.091 seconds to 0.125 seconds. The execution time is faster when using GPU, with a range of 0.042 seconds to 0.081 seconds. The difference in execution time on both CPU and GPU is because each image has a different size; therefore, it affects the execution time. The bigger the image size, hence the execution time will be longer.

TABLE IV. PREDICTION RESULT AND EXECUTION TIME OF NASNETMOBILE

Test Images	Prediction	Execution Time	
		CPU	GPU
Unripe/Green	Yellowish Green	0.157 s	0.055 s
Yellowish Green	Yellowish Green	0.150 s	0.053 s
Mid-Ripen	Mid-Ripen	0.175 s	0.062 s
Overripe	Overripe	0.189 s	0.083 s

The results in Table 4 show that the NASNetMobile cannot correctly predict the unripe/green banana, but the rest are correctly predicted due to the lower accuracy of NASNetMobile. When the prediction is executed using CPU, the execution time is in the range of 0.150 seconds to 0.189 seconds, while the execution time when using GPU is in the range of 0.053 seconds to 0.083 seconds. Overall, the execution time is faster when using the GPU. However, the execution time using MobileNet V2 is faster than NASNetMobile because the MobileNet V2 model is smaller than the NASNetMobile.

V. CONCLUSION

In this study, two pre-trained models, MobileNet V2 and NASNetMobile, were used to classify banana ripeness. The banana ripeness is divided into four classes: unripe/green, yellowish-green, mid-ripen, and overripe. The transfer learning by fine-tuning approach was applied to train both models, using different epochs and starting layers for unfreezing the model. Image preprocessing, such as using the bilateral filter, was used to remove the image's noise before training. Data augmentation such as horizontal flip, vertical flip, brightness, zoom, shear, rotation, and shifting were applied to add variations for the training data. The experiment results show that the MobileNet V2 achieves higher performance than the NASNetMobile, with the highest accuracy achieved by the MobileNet V2 is 96.18%. Choosing the starting layer for unfreezing the model will affect the performance of the model. The experiments show that choosing a higher layer as the starting layer tends to lower each model's performance. The result in each experiment also shows that increasing the epochs in training will improve the model's performance. In terms of execution time, the MobileNet V2 is faster than the NASNetMobile. The image size will affect the execution time; the more significant the image will make the execution time longer. Future research is to train the models with more data with variations of bananas in an image, applying object detection that can automatically detect a banana in an image or video stream and combines it with a classifier to create a real-time banana ripeness classifier.

ACKNOWLEDGMENT

The authors would like to convey gratitude to Universitas Atma Jaya Yogyakarta, Yogyakarta, Indonesia, for funding this research.

REFERENCES

[1] J. Naranjo-Torres, M. Mora, R. Hernández-García, R. J. Barrientos, C. Fredes, and A. Valenzuela, "A review of convolutional neural network applied to fruit image processing," Appl. Sci., vol. 10, no. 10, 2020.

[2] M. Khojastehnazhand, V. Mohammadi, and S. Minaei, "Maturity detection and volume estimation of apricot using image processing technique," Sci. Hortic. (Amsterdam)., vol. 251, no. January, pp. 247–251, 2019.

[3] A. Koirala, K. B. Walsh, Z. Wang, and C. McCarthy, "Deep learning for real-time fruit detection and orchard fruit load estimation: benchmarking of 'MangoYOLO,'" Precision Agriculture, vol. 20, no. 6. pp. 1107–1135, 2019.

[4] 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.

[5] W. Castro, J. Oblitas, M. De-La-Torre, C. Cotrina, K. Bazan, and H. Avila-George, "Classification of Cape Gooseberry Fruit According to its Level of Ripeness Using Machine Learning Techniques and Different Color Spaces," IEEE Access, vol. 7, pp. 27389–27400, 2019.

[6] M. P. Arakeri and Lakshmana, "Computer Vision Based Fruit Grading System for Quality Evaluation of Tomato in Agriculture industry," Procedia Comput. Sci., vol. 79, pp. 426–433, 2016.

[7] A. Wajid, N. K. Singh, P. Junjun, and M. A. Mughal, "Recognition of ripe, unripe and scaled condition of orange citrus based on decision tree classification," 2018 Int. Conf. Comput. Math. Eng. Technol. Inven. Innov. Integr. Socioecon. Dev. iCoMET 2018 - Proc., vol. 2018-Janua, pp. 1–4, 2018.

[8] A. Bashir, S. Suhel, A. Azwardi, D. P. Atiyatna, I. Hamidi, and N. Adnan, "The Causality Between Agriculture, Industry, and Economic Growth: Evidence from Indonesia," Etikonomi, vol. 18, no. 2, pp. 155–168, 2019.

[9] I. B. Suban, A. Paramartha, M. Fortwonatus, and A. J. Santoso, "Identification the Maturity Level of Carica Papaya Using the K-Nearest Neighbor," J. Phys. Conf. Ser., vol. 1577, no. 1, 2020.

[10] J. Pardede, M. G. Husada, A. N. Hermana, and S. A. Rumapea, "Fruit Ripeness Based on RGB, HSV, HSL, L∗a∗b∗ Color Feature Using SVM," 2019 Int. Conf. Comput. Sci. Inf. Technol. ICoSNIKOM 2019, 2019.

[11] F. M. A. Mazen and A. A. Nashat, "Ripeness Classification of Bananas Using an Artificial Neural Network," Arab. J. Sci. Eng., vol. 44, no. 8, pp. 6901–6910, Aug. 2019.

[12] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., pp. 4510–4520, 2018.

[13] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, "Learning Transferable Architectures for Scalable Image Recognition," Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., pp. 8697–8710, 2018.

[14] R. Patel and A. Chaware, "Transfer Learning with Fine-Tuned MobileNetV2 for Diabetic Retinopathy," in 2020 International Conference for Emerging Technology (INCET), 2020, pp. 1–4.

[15] K. Kangune, V. Kulkarni, and P. Kosamkar, "Grapes Ripeness Estimation using Convolutional Neural network and Support Vector Machine," in 2019 Global Conference for Advancement in Technology (GCAT), 2019, pp. 1–5.

[16] Z. Ibrahim, N. Sabri, and D. Isa, "Palm oil fresh fruit bunch ripeness grading recognition using convolutional neural network," J. Telecommun. Electron. Comput. Eng., vol. 10, no. 3–2, pp. 109–113, 2018.

[17] M. Momeny, A. Jahanbakhshi, K. Jafarnezhad, and Y. D. Zhang, "Accurate classification of cherry fruit using deep CNN based on hybrid pooling approach," Postharvest Biol. Technol., vol. 166, no. December 2019, p. 111204, 2020.

[18] R. Thakur, G. Suryawanshi, H. Patel, and J. Sangoi, "An Innovative Approach for Fruit Ripeness Classification," Proc. Int. Conf. Intell. Comput. Control Syst. ICICCS 2020, no. Iciccs, pp. 550–554, 2020.

[19] DSCEteamD, "BananaCo_images." [Online]. Available: https://github.com/DSCEteamD. [Accessed: 02-Nov-2020].

[20] D. Zhang, J. Kang, L. Xun, and Y. Huang, "Hyperspectral Image Classification Using Spatial and Edge Features Based on Deep Learning," Int. J. Pattern Recognit. Artif. Intell., vol. 33, no. 9, 2019.

[21] N. K. Manaswi, Deep Learning with Applications Using Python. Karnataka: Apress, 2018.

[22] P. Szymak, P. Piskur, and K. Naus, "The effectiveness of using a pretrained deep learning neural networks for object classification in underwater video," Remote Sens., vol. 12, no. 18, pp. 1–19, 2020.

[23] A. Rosebrock, Deep Learning for Computer Vision with Python Practitioner Bundle. Maryland: PyImageSearch, 2017.