International Journal on Advanced Science Engineering Information Technology

Vol.9 (2019) No. 6
ISSN: 2088-5334

Quality Assessment and Prediction of Philippine Mangoes: A Convolutional Neural Network Approach

Carlos Matthew P. Cases#1, Annamitz A. Rapliza#2, Francisco Emmanuel T. Munsayac Jr. III#3, Nilo T. Bugtai#4, Robert Kerwin D. Billiones#5, Renann G. Baldovino#6

#Gokongwei College of Engineering, De La Salle University - Manila, 2401 Taft Ave., Manila, 0922, Philippines
E-mail: 1carlos_matthew_cases@dlsu.edu.ph, 2annamitz_rapliza@dlsu.edu.ph, 3francisco_munsayacjriii@dlsu.edu.ph, 4nilo.bugtai@dlsu.edu.ph, 5robert.billones@dlsu.edu.ph, 6renann.baldovino@dlsu.edu.ph

***Abstract*— Philippines is one of the world's leading exporter of mangoes. The country produces many varieties of mangoes, one of which is the 'Carabao' mango. Several metric tons of mangoes are produced, and these have to be checked for defects before entering the market. With recent advances in technology, it has become efficient and relatively easy to use for these applications. The objective of this paper is to present a non-destructive method to check the quality of mangoes using computer vision (CV) and convolutional neural network (CNN) with a minimal number of samples. An experimental setup was created to simulate a production line. A webcam was used for capturing images of the mangoes, while a mini computer was used for controlling the peripherals. As basis for categorizing the mangoes as either good or bad, the Philippine National Standard (PNS) for mangoes was used. A basic background subtraction algorithm was used to extract the mango's image. With these extracted images, a 2-category network was trained, and the achieved classification accuracy was 97.21%. The goal of having a high accuracy in classifying mangoes was achieved. There are multiple paths to explore in the future, including additional feature extraction methods, different neural networks, and hardware improvements, in order to speed up the sorting process. Moreover, it may be necessary to be able to identify mangoes with only slight defects to be used for other products, such as dried mangoes, to reduce product wastage.**

***Keywords*— computer vision; convolutional neural network; mango; sorting**

I. INTRODUCTION

Mango exports are led by a number of developing countries in the world, and Philippines is one of them [1]. The leading specie of mango being exported is the 'Carabao' Mango. This mango makes up around 81.4% of the total mango production in the first quarter of 2019 [2]. Standards were made to ensure the quality of these export products, in order to further increase the marketability of these mangoes. The Philippines uses the Philippine National Standard for mangoes to determine if it is market-worthy [3]. There are ways of checking the quality of these mangoes just by its outside appearance. Fig 1 shows an example of a color scale being used for quality control of mangoes in a company.

Aside from the color of the skin, certain imperfections are also being monitored, such as the ones shown in Figure 2.

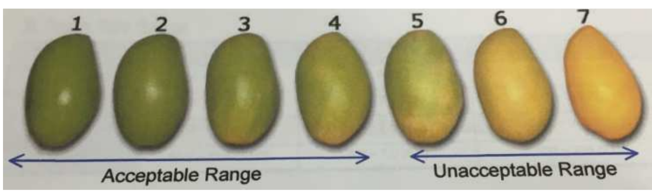

Fig. 1 Mango skin color scale

Checking for defects manually is possible, but as technology improves, more efficient ways of quality control are introduced. It is also possible to use a network of sensors to determine the quality of the mango [4]–[6], but it can be costly. Table 1 provides some examples of these defects and damages as defined in the standards.

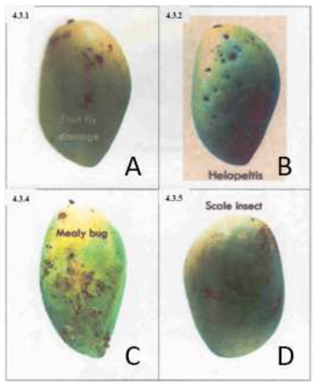

Fig. 2 Mangoes considered damaged: (A) fruit fly damage, (B) helopeltis damage, (C) mealy bug damage and (D) scale insect damage [3]

Also, some sensors may require additional setups which cannot or otherwise be difficult to be placed in an existing

TABLE I
EXAMPLES OF DEFECTS PRESENT IN 'BAD' MANGOES

Defect	Description
Discoloration	Distinct deviation from the typical color of the fruit
Heat injury	A portion of the peel which exhibits dull yellow to yellow in color
"Intul-tol"	A disorder characterized by dark brown to black depression on the peel of the fruit, not localized and already apparent even while the fruit is on the tree
Mottling	Colored spots, blotches or clouding on the peel of the fruit
Wind scar	Brownish streak, slightly elevated due to mechanical abrasion
Sooty mold	Black powdery substance appearing as irregular spots on the surface, usually on the pedicel end
Helopeltis damage	Feeding points of the insect produce corky spots sometimes randomly scattered over the fruit surface; only the skin is affected; feeding injury is often called "kurikong" or "armalite"
Insect and animal injury	Punctures, feeding and scratch scars, oviposition, entry/exit holes of insects visible to the naked eye
Mealy bug damage	Stains the fruit white due to white flour-like substance, which covers its body surface. Damaged parts are also usually covered with black sooty mold growing on the honeydew produced by the mealy bug
Scale insect damage	Feeding punctures left by the scale insect resulting in whitish to yellowish spots on the peel

production line [7]–[10]. Thus, other techniques are more recommended for this application.

As more techniques in the soft computing fields emerge, applications of these also become diverse. The use of artificial intelligence (AI), or more specifically neural networks, is dynamically growing and one if its strengths is classification of images in real time using computer vision combined with machine learning in fruit quality control. It has been previously used with orchards, bananas, and more [11]–[14]. This paper proposed a simple way in classifying good or bad mangoes, according to Philippine standards which was mentioned above. The mango is said to be good if it is free from any foreign matter, diseases, insects, and injuries.

The goal of this paper is to present a method to automate the mango sorting process with minimal input data. Convolutional neural network (CNN) was used to classify the mangoes. For the machine vision, only a webcam, a desktop computer for training, and a Raspberry Pi computer for a compact computer to interface the machine to the neural network, were used.

A. Related works

There are multiple successful attempts in integrating computer vision (CV) to industrial applications, such as fruit harvesting. One of the closest studies available is the study of Nandi et al. [15]. They used charge couple device (CCD) camera. Instead of using CNN, they used a combination of support vector regression (SVR) and multi-attribute decision-making (MADM). The system predicts the 'actual days-to-rot' parameter by using the aforementioned techniques.

Another way of counting fruits automatically is proposed by Liu et al. [16]. Instead of using high cost materials and a network of sensors such as LiDAR, depth sensor, global positioning inertial navigation systems (GPS/INS), they only used a camera with the aid of semantic structure-from-motion (SfM). SfM is an image processing approach to reconstructing objects from a single camera with different perspectives or a set of 2-D images [17]. Also, CNNs were used to detect tree trunks and fruits. The results were comparable to techniques previously used in fruit counting with expensive sensors.

The detection of apple fruits is a topic covered by another paper [18]. Their proposed method uses simple linear iterative clustering (SLIC) in order to slice an image into bigger pixels. Their method claims to differentiate incompletely red apples from completely red apples, which was a problem for previous works. The shape of the object was also taken into account to estimate the distance from the fruit to the fruit-picking robot. The technique used was slightly better than the conventional faster region-based convolutional neural network (RCNN), but is slower and less robust to noise. A similar approach is done for measuring the canopy size of crops [19]. Instead of returning the redness, this study gives the measurement of the canopy of certain crops for harvest. With the number of pixels, a mathematical model was used to approximate the crop canopy size.

Without the use of CV, detection of pests can also be possible with acoustic sensors. Another study [20] was able to successfully identify pests residing in rice storage. This may be applicable for mango pests, such as mealy bugs and scale insects, but more research needs to be done in order to achieve this.

B. Convolutional Neural Network (CNN)

CNN is one of the most widely used deep learning algorithm for classification of objects. This is a type of unsupervised learning, meaning that manual feature extraction from the images would be unnecessary [21]. Figure 3 is one of the more commonly used architectures for CNN.

Convolution takes the image and filters it several times using convolutional filters to activate certain features of the image. The activation layer enables a more efficient way to train the network by mapping the output of the convolutional layer using certain functions to carry on only the features activated to the next layer. Pooling simplifies all of these by performing downsampling, which effectively reduces the features needed to learn by the network. This can be repeated many times until the desired outcome is achieved. After learning the network learns the features, the network proceeds to classify the input. The final layers contain the classes, and the probabilities of the inputs belonging to each class [21]. The disadvantage of CNN is that it usually needs a database which may contain up to thousands of images to be trained correctly. Training usually takes up a lot of time,

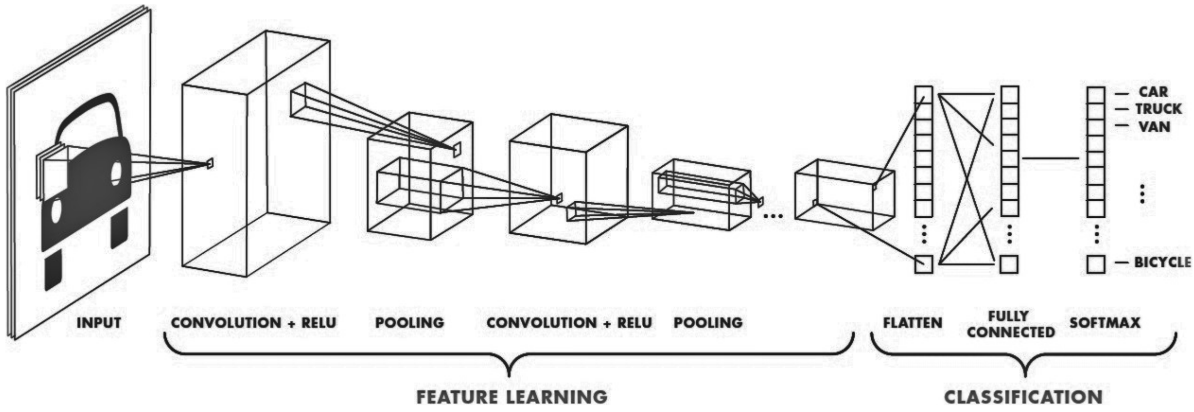

Fig. 3 CNN visualization [21]

which may last up to a few weeks. However, once the network is trained and a model is generated, real-time application is already possible. In this paper, foreground detection was used to separate the background and isolate the mango from the image.

II. Materials and Methods

Mangoes were initially acquired and manually sorted. Around 200 mangoes were available for testing. Each was checked for defects, and documented initially. The standards [3] served as the basis for 'good' and 'bad' mangoes. The experimental setup block diagram is shown in Figure 4.

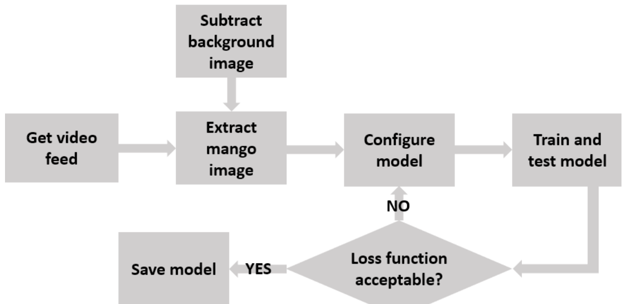

Fig. 4 Experiment flow block diagram

In the study, there was a total of 2 setups used. The first setup was used for pure training only. This consisted of a blue box with a roller below to enable the mango to show the whole fruit. For the sorting machine, solenoids and rollers were used to block the mangoes, and to roll the mango and get a full view for the camera that will be placed on top of the assembly. A motor attempts to turn the mango roughly 120 degrees on an axis and stops, and the camera takes a photo of it. This is repeated 3 times or more, until the mango is fully viewed. The rolling and the image taking takes about 3 seconds per mango to complete. Fig 5 shows a sample photo in the setup.

The place where the mangoes will be placed is well-lit by artificial light sources so the mangoes would not be affected by changes of the ambient lighting. The feed from the camera goes to a desktop using OpenCV. This was also used to subtract the background to obtain only the mango image. A blue background was used for ease of foreground detection. For a stationary camera, this technique is reliable. Basically, an image of the background is taken, and it is subtracted to only obtain the foreground. In this case, the foreground is the mango itself. The image seen from the computer's perspective is depicted by Fig. 6.

The OpenCV Python library interfaces the camera feed to the computer in preparation for the machine learning algorithm. Keras was used for the learning algorithm. This is a CNN-based machine learning algorithm which can be trained with minimum data without overfitting [22]. The activation function used for this application was rectified linear unit (ReLU). The equation $y = \max(0, x)$ describes the function and Figure 7 displays the graph. It can be seen that negative values are automatically mapped to 0, meaning that these values will have no effect to the succeeding neurons.

The program flow for the first 2 program versions is as follows:

Data: Mango images
Result: CNN Model
1 Initialize mango image location
2 Initialize number of samples and epochs
3 Specify number training and testing samples
4 Adjust image dimensions for CNN
5 Select random images to transform (e.g. flip) to pad data
6 Initialize CNN model parameters
7 Train neural network
8 Test neural network

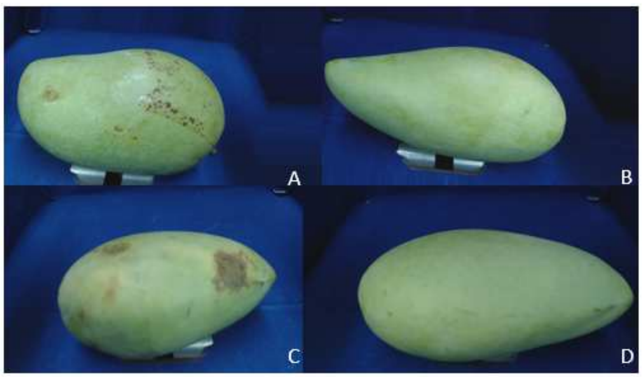

Fig. 5 Mangoes in initial setup: (A and C) 'bad', (B and D) 'good' mango

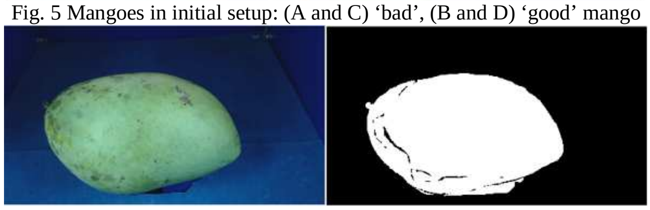

Fig. 6 Foreground detection

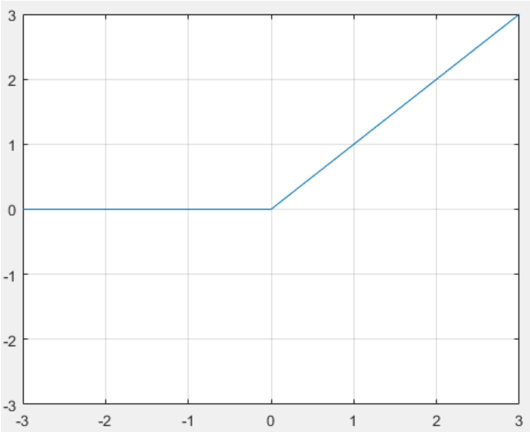

Fig. 7 ReLU graph

However, for the rest of the program versions, line 5 is omitted. This was not needed anymore since more pictures of mangoes were taken at this point.

There are multiple iterations to the program. Since the goal was to train with minimal data, multiple snapshots were taken for each mango. In the first iteration of the program, the data was randomly transformed in order to increase the data. This led to 2,000 samples for training, and 800 for testing, and performed in 50 epochs. For the 2nd iteration of the program, the number of samples was increased from 2,000 to 3,000 images to see if it would improve the accuracy of the network. The next iteration did not use randomly transformed images. Instead, it took 2,000 images for training, and 800 images for testing. The number of epochs was also decreased from 50 to 1. From this, the 4th iteration had 5 epochs instead of 1. The results were satisfactory, so it was recommended to move to larger set of mangoes. More mangoes were added with different defects on the 5th iteration. On the 6th iteration, a new hardware setup was introduced. In Figure 8, a more sophisticated machine was used to mimic industrial applications.

This consists of a conveyor, solenoids, a camera, and a Raspberry Pi computer. The conveyor's purpose is to move the mangoes forward, while the solenoids would block the mangoes. The solenoids have rollers on their ends to turn the mango for the mango viewing process. A total of 4 solenoids, 2 for queuing the mangoes, and the next 2 solenoids were for sorting the mangoes. This was still run in 5 epochs.

The 7th iteration had the number of epochs changed from

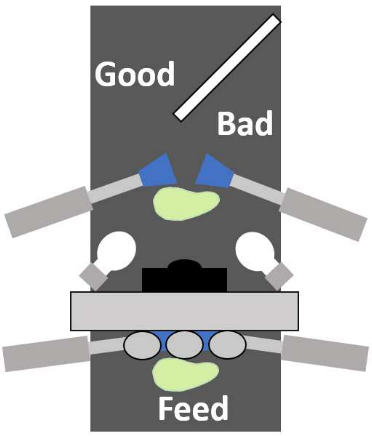

Fig. 8 New hardware setup (top view)

5 to 20 in the hopes of increasing the accuracy. In the 8th iteration, the number of epochs was again increased from 20 to 30. The image databases were combined in the 9th iteration to widen the scope of the neural network. There were a total of 4,150 samples for training and 1650 samples for testing. This database also included images with the background only. Finally, the last program utilizes every available image in the database. There are also added images of mangoes with artificial defects to simulate worse conditions of mangoes. The artificially damaged mango is in Figure 9.

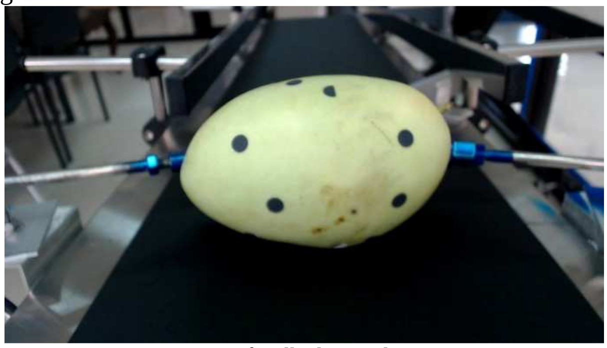

Fig. 9 Artificially damaged mango

This final model was saved, and uploaded into the Raspberry Pi minicomputer in order to scale down the size of the desktop. The program flow is as follows: roll the mango, capture images, extract the image(s), classify, and activate the appropriate solenoid. A sample image is exhibited in Figure 10.

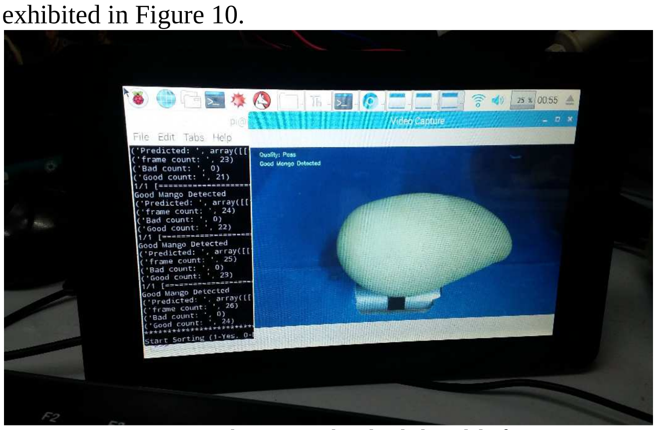

Fig. 10 Raspberry Pi with uploaded model of CNN.

III. Results and Discussion

The summary of results of each iteration are documented in Table 2. Multiple iterations were made to improve the accuracy, and also to update the dataset when it was transferred to the sorting machine. It can be inferred that a very high epoch is not usually needed. Since one of the strengths of Keras is deep learning with a small pool of data, the training data only had 2,000 images, with equal number of good and bad mangoes, fed to it. This resulted to 1,000 images of 'good' mangoes, and 1,000 images of 'bad' mangoes. To increase this number further, the images were randomly transformed. This achieved a classification accuracy of 89.55% for training, and 95.38% for testing. Note that this was achieved with 50 epochs. Next, it was investigated if the number of samples would increase the testing accuracy. However, it decreased to 94.25%.

TABLE II
RESULTS SUMMARY

Iteration	Epochs	Training			Testing		
		Sample	Loss Function	Accuracy	Sample	Loss Function	Accuracy
1	50	2000	0.3108	89.55%	800	0.1758	95.38%
2	50	3000	0.1696	94.79%	800	0.1963	94.25%
3	1	2000	0.3573	86.40%	800	0.0101	100%
4	5	2000	0.0375	99.40%	800	0.0000155	100%
5	5	2000	0.0276	99.35%	800	0.0033	99.88%
6	5	2000	0.2225	91.30%	800	0.2357	91.25%
7	20	2000	0.0810	96.95%	800	0.0185	99.62%
8	30	2000	0.0688	98.25%	800	0.0269	99.00%
9	20	4150	0.0846	97.97%	1650	0.0513	97.61%
10	5	5550	0.1395	94.99%	2320	0.0628	97.21%

Then the number of epochs were reduced to 1, because 50 took too long to process. It was tested if it could be done faster, while not compromising classification accuracy. Also, the sample data was decreased back to 2,000. It increased the testing accuracy to 100%, but the training accuracy was lowered to 86.4%. For a better fit to the data, the number of epochs was increased to 5, and the results were ideal. For the 5th iteration, new mango images were introduced to the database in order to see the effect of it. It did not affect the classifying accuracy of the network significantly.

The camera was moved to a new environment, resulting to a new set of images. Iteration 6 had a lower accuracy than the previous test, with only 91.3% for training and 91.25% for testing. Changing the number of epochs to 20 further increased its accuracy to 99.62%, and further increasing the epochs to 30 did not improve the results significantly. For the next iteration, the image database of the test setup and the sorting machine setup were combined. It also included pictures with no mangoes. Lastly, the network was tested if it would classify a mango as 'bad' with artificial defects. This was to simulate the conditions of a worse mango. It achieved an accuracy of 94.99% in training, and 97.21% in testing. This was achieved in only 5 epochs.

Generally, the accuracy increases with increased number of epochs. However, it only increases from epochs 1-5, and the loss function does not significantly decrease with epochs greater than 5. A large loss function increase was seen in iteration 6, which is where the new setup was introduced to the system. It had to take more images with the more advanced machine before the accuracy increased. Additionally, introducing newer samples decreased the overall accuracy of the neural network, as seen in iterations 8-10. But, from iterations 3 and 4, it can be seen that a 100% accuracy was achieved. This was because the mangoes introduced to the system were visibly good or bad, and the setup was ideal for image processing. The mango had a plain blue background, which makes the setup free from external factors, such as ambient lighting and shadows. The classification rates are still acceptable, which were greater than 90%. Per epoch, the time it took to train the network on 2,000 samples was, on average, 156 seconds. Extrapolating from the time it takes on average to classify a single mango, the setup can sort up to 870 mangoes per hour.

IV. CONCLUSION

The researchers were able to achieve a relatively high accuracy of 97.21% using only a few mangoes. This was obtained with the use of Keras, a high-level neural network library implemented in Python. The classified mangoes are Philippine National Standard [3] compliant, which caters to both domestic and export markets. Future directives may include classifying the defect. This system only detects good and bad mangoes, and some markets accept mangoes with defects, but only up to a certain extent. There are different tolerances for different defects, and this may increase the number of mangoes classified as good enough to be sold in the market. This should lead to more mangoes sold.

This study achieved automated mango sorting. It was implemented minimally, through CNN and a webcam. The hardware can be upgraded for faster sorting of mangoes. For example, the computer can be changed into a more powerful one. The camera settings can be adjusted so it can take pictures while the mango is being rotated without blurriness. The conveyor can also have stronger motors to accommodate more mangoes, and to push mangoes faster.

Other techniques of classification can be investigated to decrease training time for less downtime, or faster classification time without compromising accuracy. Support vector machine (SVM) can be tested, or other CNN techniques, such as Fast Region-based CNN (R-CNN). Data pre-processing can also be applied to make feature learning easier for the CNN. There are other types of foreground detection, such as the Gaussian Mixture Model (GMM). These parameters can be tested to speed up mango sorting.

ACKNOWLEDGMENT

The researchers would like to thank the Engineering Research and Development for Technology (ERDT) of the Department of Science and Technology (DOST) and the

Philippine Center for Postharvest Development & Mechanization (PHILMECH) for the research funding and support for publishing the research.

REFERENCES

[1] K. Stark, V. Couto, and G. Gary, “The Philippines in the Mango Global Value Chain,” 2017.

[2] PSA, “Major Fruit Crops Quarterly Bulletin,” Philippine Statistics Authority, 2019. [Online]. Available: http://www.psa.gov.ph/fruits-crops-bulletin/mango. [Accessed: 17-Jul-2019].

[3] D. of T. and I. Philippines, “Philippine National Standard: Fresh Fruit - Mangoes,” vol. 2015, 2004.

[4] L. Y. Chen et al., “Development of an electronic-nose system for fruit maturity and quality monitoring,” Proc. 4th IEEE Int. Conf. Appl. Syst. Innov. 2018, ICASI 2018, pp. 1129–1130, 2018.

[5] Y. Hasegawa, A. L. Spetz, and D. Puglisi, “Ethylene gas sensor for evaluating postharvest ripening of fruit,” 2017 IEEE 6th Glob. Conf. Consum. Electron. GCCE 2017, vol. 2017-January, no. Gcce, pp. 1–4, 2017.

[6] N. H. Hasanuddin et al., “Metal oxide based surface acoustic wave sensors for fruits maturity detection,” 2016 3rd Int. Conf. Electron. Des. ICED 2016, no. 1, pp. 52–55, 2017.

[7] W. Lang and R. Jedermann, “What Can MEMS Do for Logistics of Food? Intelligent Container Technologies: A Review,” IEEE Sens. J., vol. 16, no. 18, pp. 6810–6818, 2016.

[8] E. Vitzrabin and Y. Edan, “Changing Task Objectives for Improved Sweet Pepper Detection for Robotic Harvesting,” IEEE Robot. Autom. Lett., vol. 1, no. 1, pp. 578–584, 2016.

[9] P. Leekul, S. Chivapreecha, C. Phongcharoenpanich, and M. Krairiksh, “Rician k-Factors-Based Sensor for Fruit Classification by Maturity Stage,” IEEE Sens. J., vol. 16, no. 17, pp. 6559–6565, 2016.

[10] P. Leekul and M. Krairiksh, “A sensor for continuous fruit classification using Rician k-factor,” 2018 Int. Symp. Antennas Propag., pp. 1–2, 2018.

[11] S. Bargoti and J. Underwood, “Deep fruit detection in orchards,” in 2017 IEEE International Conference on Robotics and Automation (ICRA), 2017, pp. 3626–3633.

[12] S. Marimuthu and S. Mohamed Mansoor Roomi, “Particle Swarm Optimized Fuzzy Model for the Classification of Banana Ripeness,” IEEE Sens. J., vol. 17, no. 15, pp. 4903–4915, 2017.

[13] Y. Polinar, K. F. Yaptenco, E. K. Peralta, and J. U. Agravante, “Near-infrared spectroscopy for non-destructive prediction of maturity and eating quality of ‘Carabao’ mango (Mangifera indica L.) fruit,” Agric. Eng. Int. CIGR J., vol. 21, no. 1, pp. 209–219, Apr. 2019.

[14] K. Sujatha, R. S. Ponmagal, V. Srividhya, and T. Godhavari, “Feature extraction for ethylene gas measurement for ripening fruits,” Int. Conf. Electr. Electron. Optim. Tech. ICEEOT 2016, pp. 3804–3808, 2016.

[15] C. S. Nandi, B. Tudu, and C. Koley, “A Machine Vision Technique for Grading of Harvested Mangoes Based on Maturity and Quality,” IEEE Sens. J., vol. 16, no. 16, pp. 6387–6396, Aug. 2016.

[16] X. Liu et al., “Monocular Camera Based Fruit Counting and Mapping With Semantic Data Association,” IEEE Robot. Autom. Lett., vol. 4, no. 3, pp. 2296–2303, Jul. 2019.

[17] “Structure From Motion From Multiple Views.” [Online]. Available: https://www.mathworks.com/help/vision/examples/structure-from-motion-from-multiple-views.html. [Accessed: 20-Jul-2019].

[18] X. Liu, D. Zhao, W. Jia, W. Ji, and Y. Sun, “A Detection Method for Apple Fruits Based on Color and Shape Features,” IEEE Access, vol. 7, pp. 67923–67933, 2019.

[19] X. A. P. Calangian *et al.*, “Vision-based Canopy Area Measurements,” in 2018 IEEE 10th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM), 2018, pp. 1–4.

[20] R. M. C. Santiago, S. L. Rabano, R. K. D. Billones, E. J. Calilung, E. Sybingco, and E. P. Dadios, “Insect detection and monitoring in stored grains using MFCCs and artificial neural network,” in TENCON 2017 - 2017 IEEE Region 10 Conference, 2017, pp. 2542–2547.

[21] “Convolutional Neural Network - MATLAB & Simulink.” [Online]. Available: https://www.mathworks.com/solutions/deep-learning/convolutional-neural-network.html. [Accessed: 20-Jul-2019].

[22] Chollet, F. (2015) Keras, GitHub. https://github.com/fchollet/keras