IAES International Journal of Artificial Intelligence (IJ-AI)
Vol. 11, No. 4, December 2022, pp. 1353~1361
ISSN: 2252-8938, DOI: 10.11591/ijai.v11.i4.pp1353-1361

Classification of jackfruit and cempedak using convolutional neural network and transfer learning

Putra Sumari[1], Azleena Mohd Kassim[1], Song-Quan Ong[2], Gomesh Nair[1], Al Dabbagh Ragheed[1], Nur Farihah Aminuddin[1]
[1]School of Computer Sciences, Universiti Sains Malaysia, Penang, Malaysia
[2]Institute for Tropical Biology and Conservation, Universiti Malaysia Sabah, Kota Kinabalu. Sabah, Malaysia

Article Info

Article history:

Received Sep 3, 2021
Revised Jun 24, 2022
Accepted Jul 23, 2022

Keywords:

Cempedak
Computer vision
Deep learning
Jackfruit
Machine learning
Neural network

ABSTRACT

Jackfruit (Artocarpus integer) and Cempedak (Artocarpus heterophyllus) are two different Southeast Asian fruit species from the same genus that are quite similar in their external appearance, therefore, sometimes difficult to be recognized visually by humans, especially in the form of pictures. Convolutional neural networks (CNN) and transfer learning can provide an excellent solution to recognize fruits, where the methods are known to be able to classify objects with high accuracy. In this study, several models were proposed and constructed to recognize the Jackfruit and Cempedak using a deep convolutional neural network (DCNN). We proposed our custom-made own CNN model and modify five transfer learning models on pre-trained VGG16, VGG19, Xception, ResNet50, and InceptionV3. The experiment used our own dataset and the result showed that the proposed CNN architecture was able to provide an accuracy between 89% to 93.67% compared to the other CNN transfer learning.

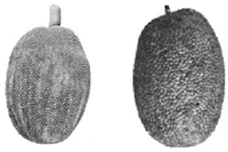
Corresponding Author:

Azleena Mohd Kassim
School of Computer Sciences, Universiti Sains Malaysia
Penang, Malaysia
Email: azleena.mk@usm.my

1. INTRODUCTION

The Jackfruit (*Artocarpus heterophyllus*) and Cempedak (*Artocarpus integer*) are tropical fruits commonly found in the Southeast Asia regions. The sample image of Jackfruit is shown in Figure 1(a) while the sample images of Cempedak is shown in Figure 1(b) [1]. Both fruits belong to the genus *Artocarpus* and the mulberry tree family *Moracceae*, which shows its characteristics in irregular oval and slightly curvy shape, in addition to its large size, although Cempedak is known to have a somewhat cylindrically shape [1].

The Cempedak skin turns yellow when it is ripe or old, whereas the Jackfruit usually retains its green-coloured skin but may also turn yellowish or brownish in certain cases. When looking at both of these fruits from a distance, the distinction is very difficult, thus it may be easier to tell them apart by looking at them closely with close attention. However, the outward appearance of the fruit makes for a distinct challenge. Both fruits produce big compound cauliflorous fruits but the Cempedak is rather small when compared to the Jackfruit [2], and has a thinner peduncle. Jackfruit may range in size from 20 cm to 90 cm long and 15 cm to 50 cm wide, with weights ranging from 4.5 kg to 50 kg or even more. When the fruit matures, the 'skin' or exterior of the compound or bundled fruit, is green or yellow, with many hard, conical points linked to a thick, rubbery light yellow or white wall [3]. As for the tree, the bark may have a greyish-brown colour where white gummy latex can be emitted if the tree is injured, and the leaves have a somewhat rough feel [4].



(a) (b)

Figure 1. Images of; (a) Jackfruit (Artocarpus heterophyllus), (b) Cempedak (Artocarpus integer) [1]

Cempedak range in size from 10 cm to 15 cm wide and 20 cm to 35 cm long, and can be cylindrical or oval. The thin, leathery skin is greenish, yellowish, or brown in hue, and has pentagons with elevated bumps or flattened eye sides [2]. For the Cempedak tree, the stem appears to be rounded and the bark appears to be in greyish-brown to dark brown colour, where it also emits a latex when injured but appears in a more milky form [4]. Odour identification and texture of fruit bundles are the most common approach to distinguish between Jackfruit and Cempedak, in which Cempedak usually exhibit a stronger smell and softer texture [1]. Commonly, people mistake Jackfruit with Cempedak and vice versa based on the size and sometimes the odour. However, from the naked eye, it is often deceiving to differentiate these fruits, especially when the fruits are represented in the form of images. For images, the recognition and differentiation of these two fruits rely on the visual and size factors, and odour would not be handy. Thus, to cater to this issue, the idea to distinguish between both fruits using deep convolutional neural networks (DCNN) and transfer learning algorithms is proposed in this paper.

Methods for quality assessment and automated harvesting of fruits and vegetables have been explored by many researchers, but the latest technologies have been created for limited classes and small data sets. Often the application of DCNN would require a different algorithm to train the model of best fit, but there is no work to the best of found knowledge that presented results related to the accuracy of classification to distinguish between Jackfruit and Cempedak. The study presented in this paper aims to construct a custom-made DCNN classification system for Jackfruit and Cempedak fruits and compare the performance of the proposed classification with some existing transfer learning algorithms such as Xception, VGG16, VGG19, ResNet50, and InceptionV3.

2. RESEARCH BACKGROUND

Due to the similarities between classes and inconsistent features within the cultivar, fruit, and vegetable, the classification presents significant problems. Due to the wide diversity of each type of feature, the selection of appropriate data collection sensors and feature representation methods is particularly critical. There are limitations of current methods for quality assessment and automated harvesting of fruits and vegetables, especially those that have limited classes and small data sets [5]–[7]. The problem is multidimensional, with many hyper-dimensional properties, which is one of the fundamental problems in current machine learning techniques [8]–[10]. It was concluded that machine vision methods are ineffective when dealing with multi-characteristic, hyperdimensional data for classification [11], [12].

Fruits and vegetables are divided into several groups, each of which has its own set of characteristics. Due to the paucity of basic data sets, specific classification methods are limited. The majority of trials are either restricted in terms of categories or dataset size. The present study into building a pre-trained convolutional neural network (CNN) is a step toward creating the capacity to supply turnkey computer vision components. These pre-trained CNNs, on the other hand, are data-driven, and there is a scarcity of huge datasets of fruits [13].

Rahnemoonfar and Sheppard [14] utilised a deep neural network (DNN) to apply to robotic agriculture, where it focused on images of tomato fruits found on the internet. They tweaked the Inception-ResNet architecture and applied a variety of training data to train the model (under the shade, surrounded by leaves, surrounded by branches, the overlap between fruits). Their search results revealed an average test accuracy of 93% on synthetic pictures and 91% on actual photos. Tan *et al.* [15] used CNN to create a model that can notify a driver of a car when he or she is sleepy, extending it with the method known as the Staked Deep CNN. To extract features and apply them in the learning phase, the DNN was created. The CNN classifier uses the SoftMax layer to determine whether a driver is sleeping or not. Besides that, the Viola-Jones face detection method was adapted where the eye area was removed from the face when it was discovered. The Staked Deep CNN was found to overcome the drawbacks of standard CNN, such as location accuracy in regression, and had a 96.42% accuracy rate. Tan *et al.* [15] suggested that transfer learning can be used in the future to improve the performance of the model.

Based on four different varieties of fruits, a method for recognising the category of the fruit (litchi, apple, grape and lemon) was provided using photos captured using smartphones, which were then processed using a contemporary detection framework [16]. Because the model is trained using a new data set of 2403 data from four different fruit classes, CNN was utilised to train it. The model's total performance was outstanding, with a precision of 99.89%, whereby the CNN was successful in identifying the category of the fruit. The researchers planned to use the algorithm to detect more variety of fruits in the future.

CNN was applied in a work that classified the ripeness of mulberry fruit with some fine-tuning to help improve the classification's accuracy [17]. From the five CNN models used, the AlexNet and ResNet-18 networks appeared to have the best performance, with ResNet-18 showing the most superiority. Thus ResNet-18 was claimed to be a good model to be applied for precise classification for the classification of ripeness of mulberry fruit. Works that have extended the traditional CNN were reported to have found promising results. For instance, the work that presented the deep learning-based fine-tuned MobileNet CNN to classify fruits such as strawberry and cherry [18]. The accuracy level was reported to be high, which was about 98.60%. Ma *et al.* [19] stated that the deep convolutional neural networks (DCNN) method has benefits over CNN where the framework delivers a uniform feature extraction-classification. Many researchers worked on expanding and customizing the DCNN to suit the problems to be solved, as the original DCNN has limitations such as the fixed depth, fixed activation function, fixed filter size, and so on. Palakodati *et al*. [20] applied the CNN with the help of Softmax in their work of fresh and rotten fruits classification. Their proposed model showed a result that is better than the state-of-the-art methods.

Hussain *et al.* [21] presented the DCNN to solve the fruit recognition problem from 15 different categories of fruits. Since the previous techniques had issues, especially in the changes of external environments, DCNN was reported to be able to efficiently meet real-world application requirements. The researchers compared their results with existing work that applied DNN and achieved a similar accuracy level, but the advantage was that their work used more complicated datasets, closer to real-world applications [21]. The high accuracy level of CNN in fruits classification and recognition by previous works explains the popularity of the CNN algorithm in this area of work. Further improvements are constantly carried out and the custom or extended variations of CNN such as DCNN are getting more popular these days. Thus, in this paper, the use of DCNN in tropical fruit such as Jackfruit and Cempedak is further investigated.

3. RESEARCH METHOD

The overall flow of the proposed research is shown in Figure 2. Dataset was established by our own collection due to the lack of a similar dataset in existing works. The dataset was prepared using reshaping and double up using augmentation. At the same time, various CNN models are proposed. Then dataset was split for training and testing and finally, accuracy was measured. Tensorflow and Keras library specific for augmentation and data preprocessing have been used for the augmentation task.

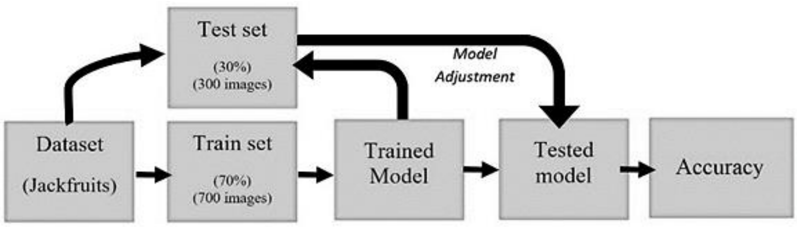


Figure 2. The overall flow of model development

3.1. Dataset

The fruit dataset is collected manual and was shot with a digital single-lens reflex (DSLR) camera (Canon 7D, • 22.3 mm × 14.9 mm complementary metal-oxide semiconductor (CMOS) sensor, red, green, blue color model (RGB) color filter array, 18 million effective pixels). The data used were divided into two classes: Cempedak (Artocarpus integer) and Jackfruit (Artocarpus heterophyllus) with a total of 1000 images (each class consisted of 500 images) with a resolution of 4608×3456 pixels. The images were collected with three spectrums of lights: green, red, blue (by introducing an external gel filter on the flashlight), and white light. The reason was to have a dataset that could represent high variability in position and number of fruits devising a real scenario. This dataset is available on https://drive.google.com/drive/folders/1z8LMNMtILWnGaxF9c-nYjJZUprx0c5uU?usp=sharing by request, to be made to putras@usm.my.

3.2. Data pre-processing

Data augmentation was applied to the dataset to double the volume of the dataset into 2000 images. During augmentation, the images were randomly rotated within 0 to 180 degrees, randomly shift into a vertical or horizontal direction, randomly shear transform, zooming to randomly scale the image into different sizes, flipping 50% horizontally, and lastly, filling up the image after augmentation like rotation or translation. Next, the entire image was reshaped to 224×224×3. In order to build the CNN model and faster convolution, the image was then converted into a NumPy array and labelled based on the two classes. Next, the images were split into training and testing sets by the ratio of 80:20. The training dataset was allocated 10% for the validation set.

3.3. Proposed convolutional neural networks

The proposed DCNN model applied for classifying the Jackfruit and Cempedak is depicted in Figure 3. Unlike the work in [20] that had 3 convolutional layers, our proposed model comprises 4 main convolutional blocks-convolution 1, 2, 3, and 4. The more convolutional blocks could extract more features for better learning. Random_uniform was employed for the kernel initializer, which allowed a uniform distribution to occur. The maximum value used for the random_uniform is 0.05, whereas -.0.05 is set as the minimum value. The size of the input image was set to 224×224×3. The shape of input and output tensors remained the same because there was no padding.

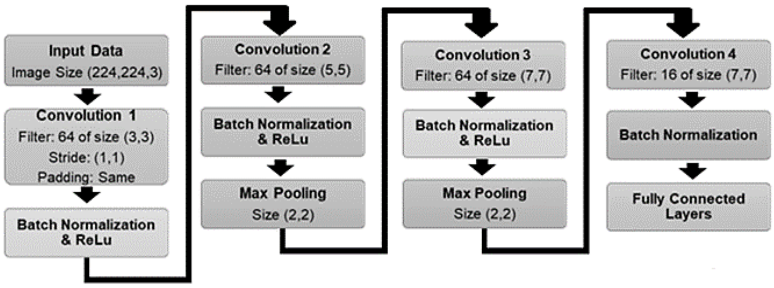


Figure 3. Proposed DCNN model

In this model, convolution 1 used 64 filters where a filter size of 3×3 was applied. Before moving on to the next layer, batch normalization was utilized. After normalizing the rectified linear activation function (ReLU), then we apply the activation function for the following convolution layer. At the end of each convolutional layer, we would have an output, where this would be an input to the max-pooling layer. The pool size of the max-pooling layer is set to 2×2. The max-pooling layer's role is to take only the positive value and discard the negative value. The negative value would be eliminated due to its unimportance to learning. The amount eliminated was almost half the original size due to the 2×2 size of the filter. In the classification, the reduction which is called downsampling would cause a decrease in the parameters number, because only the necessary features were used, which would then, help to decrease the memory size and computational time. For the dropout, the value 0.5 was used as this would ensure fast computation.

In this model, convolution 2 also has 64 filters with the kernel size of 5×5 kernels, followed by convolution 3 which has 64 filters as well, with a kernel size of 7×7. The final layer, convolution 4 has 16 filters with a kernel size of 7×7. The last is dense layer used 3 fully connected with 1000 nodes each with two classes. Categorical cross-entropy is applied as the loss function and the performances of the optimizers were compared with three levels of epochs (25, 50, and 75), with a learning rate of 0.001 based on these 2 optimizers: 1) Adam 2) stochastic gradient descent (SGD).

3.4. Transfer learning

We also used the transfer learning on VGG16, VGG19, Xception, ResNet50, and InceptionV3 to compare with our proposed DCNN model. These CNN architectures had been trained mostly on the Imagenet dataset and can classify large classes. The intention here is to tune the existing architecture for the best performance on our fruit classification.

3.4.1. VGG16

VGG16 was developed for a visual recognition challenge in the year 2014 [22]. VGG16 is composed of 13 convolutional layers, 5 max-pooling layers, and 3 fully connected layers as shown in

Figure 4. The convolutional layers and the fully connected layers are tunable, thus there are 16 tunable parameters altogether, which is how this model got its name-VGG16. The first block has 64 filters, and the number doubles in each block until the block reaches 512 filters. The number of classes at the fully connected layer is set to 2 to suit the label. Adam optimizer is selected for the learning rate and optimization.

3.4.2. VGG19

The VGG19 in Figure 4 is an upgrade to the VGG16 model. VGG19 enhances VGG16 architecture by eliminating AlexNet's flaws and increasing system accuracy [23]. It is a 19-layer convolutional neural network model and is constructed by stacking convolutions together, however, the depth of the model is limited due to a phenomenon known as diminishing gradient. Deep convolutional networks are tough to train following this issue.

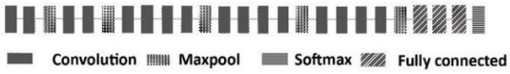


Figure 4. VGG-19 architecture

3.4.3. ResNet50

ResNet stands for Residual Network in short. We freeze the ResNet-50 [24] model consisting of 5 stages each with a convolution and identity block as shown in Figure 5. In the convolution block, there are 3 convolution layers and in each identity block, there are 3 convolution layers as well. It was trained on a million photos from the ImageNet database in 1000 categories. We changed the classes into 2 classes. The model comprises approximately 23 million trainable parameters, indicating a deep architecture that improves image identification. When compared to building a model from scratch, where usually a large amount of data must be collected and trained, using a pre-trained model is a highly effective option. ResNet-50 is a helpful tool to know because of its high generalisation performance and low error rates on recognition tasks.

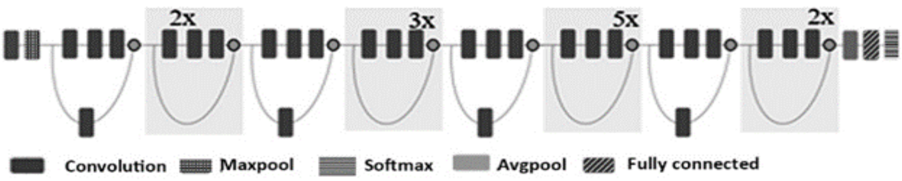


Figure 5. ResNet50 architecture

3.4.4. InceptionV3

InceptionV3 is a 48-layer deep pre-trained convolutional neural network model [25], applied in this work, as shown in Figure 6. It is a version of the network that's already been trained on over a million photos from ImageNet. It's the third version of Google's Inception CNN model, which was first proposed during the ImageNet recognition challenge. InceptionV3 is capable of categorising photos into 1000 different object types. Consequently, the network has learned a variety of rich feature representations for a variety of images. The network's picture input size used was 299×299 pixels. In the first stage, the model extracted generic features from input photos and then classified them using those features in the second portion. On the ImageNet dataset, Inception v3 has been demonstrated to achieve better than 78.1% accuracy and roughly 93.9% accuracy in the top 5 results [26].

3.4.5. Xception

Xception is an architecture that was developed by Google as shown in Figure 7. The name "Xception" comes from the term "extreme inception". In this work, we set the input as 229×299×3. The Xception has 36 convolutional layers which build the feature extraction base of the network, and these layers are then split into 14 modules. As this model is inspired by the Inception architecture, it has the same parameters numbers as the Inception V3. Xception proved to have small gains in classification performance on the ImageNet dataset [27]. In the 14 modules of Xception, each has linear residual connections, excluding the first and last modules. For the experiment, the number of classes was replaced with 2, in the last Fully Connected layer.

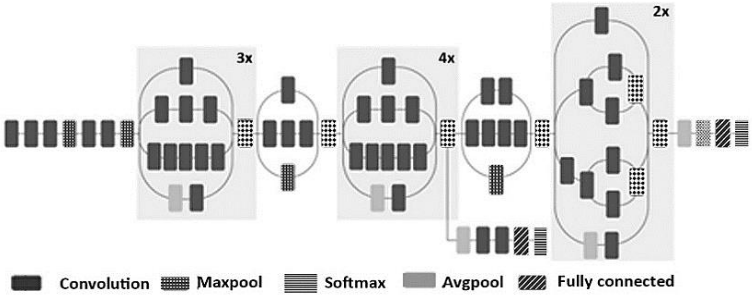


Figure 6. InceptionV3 architecture

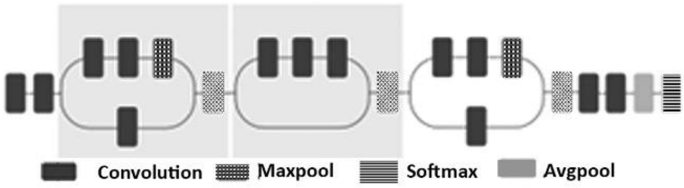


Figure 7. Xception architecture

4. RESULTS AND DISCUSSION

Table 1 shows that the proposed DCNN architecture was able to provide accuracy in 6 different epochs, with an accuracy value of 0.8933 with epoch 25 until the accuracy value of 0.9367 with epoch 75. The highest value is 0.9367 using SGD optimizer and 75 epochs. The graph to represent the comparison between the proposed method (highlighted) and other models are shown in Figures 8(a) and 8(b) respectively.

In the experiment involving the Adam optimizer, the proposed model showed the best results in the epoch values 25 and 50. However, the InceptionV3 showed the best result for the higher epoch of 75. Figure 9(a) shows the performance of the proposed model using Adam optimizer with the InceptionV3 model.

In the experiment with the SGD optimizer, the proposed model showed the best result when the lower epoch of 25 was used. Then, with the higher values of epochs, 50 and 75, the VGG16 outperformed the proposed model with the best value. Nonetheless, the proposed model was the second-best in the SGD experiment for the higher epochs. Figure 9(b) shows the performance of the proposed model using SDG optimizer with the VGG16 model.

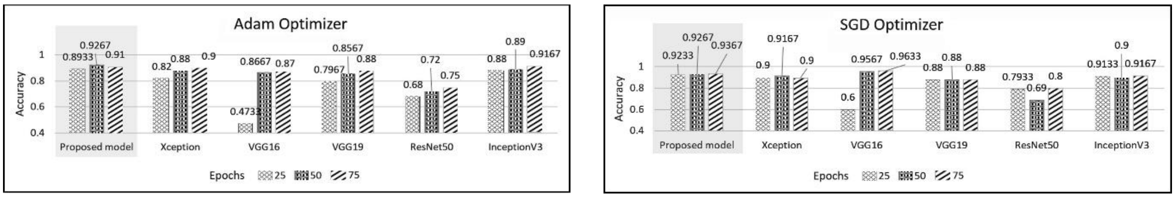
Table 1. Accuracy of the proposed DCNN and transfer learning models with Adam and SGD optimizers

Variable	Adam			SGD		
Epoch	*25*	*50*	*75*	*25*	*50*	*75*
Proposed DCNN	0.8933	0.9267	0.9100	0.9233	0.9267	0.9367
Xception	0.8200	0.8800	0.9000	0.9000	0.9167	0.9000
VGG16	0.4733	0.8667	0.8700	0.6000	0.9567	0.9633
VGG19	0.7967	0.8567	0.8800	0.8800	0.8800	0.8800
ResNet50	0.6800	0.7200	0.7500	0.7933	0.6900	0.8000
InceptionV3	0.8800	0.8900	0.9167	0.9133	0.9000	0.9167

Overall, to compare the accuracy results, it was observed that the results from the SGD optimizer were better than those from the Adam optimizer. The accuracy of the VGG16 in the SGD was the highest with an epoch value of 75. When the epoch value in SGD was high, VGG16 provided more stable and consistent performance throughout the epoch and it was evident as shown in Figure 9(b). In most of the models too, it was observed that the accuracy increased when the epoch value increased in each of the optimizers. Therefore, it showed that the higher the epoch, the higher accuracy.

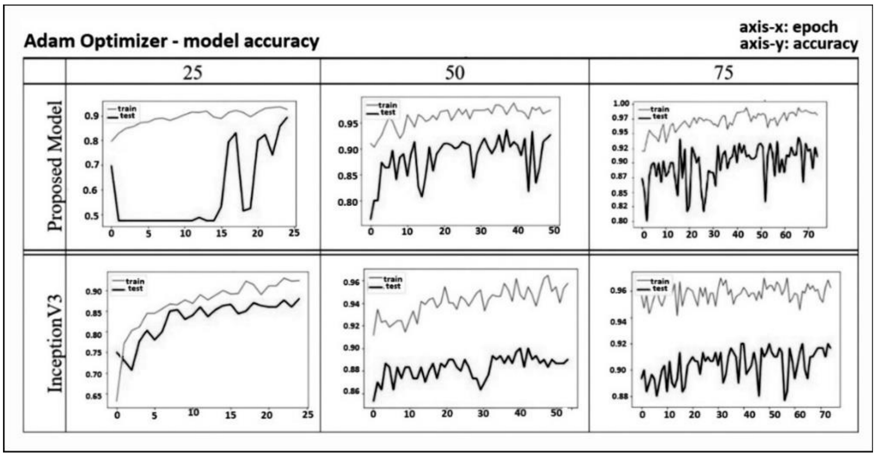
The proposed model provided the best results for epoch 25 for both Adam and SGD optimizer, which shows that the proposed model strength was at the lower epoch value. It still managed to get the best result in epoch 50 in Adam optimizer. Although for a higher epoch, the accuracy was overtaken by

InceptionV3 in Adam and by VGG16 in SGD, the proposed model still came in as the second-best after the respective models, which showed the promising performance from this proposed model generally.

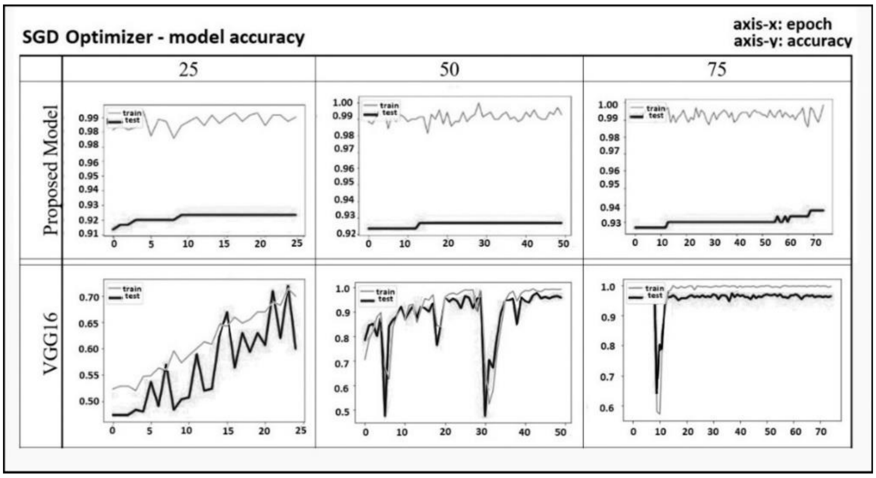


(a) (b)

Figure 8. Accuracy at three levels of epochs of the model; (a) Adam optimizer, and (b) SGD optimizer



(a)



(b)

Figure 9. Accuracy comparison based on three epochs 25, 50 and 75 between the proposed work with (a) InceptionV3 model on the use of Adam Optimizer, and (b) VGG16 model on the use of SGD Optimizer

5. CONCLUSION

Cempedak (Artocarpus heterophyllus) and Jackfruit (Artocarpus integer) are highly similar in their external appearances and are difficult to recognize visually by a human and due to the similarities between classes and inconsistent features within the cultivar, fruit, and vegetable classification, which presents significant problems. This paper proposed 6 various CNN architectures to classify Jackfruit and Cempedak. The experiment conducted on our data collection showed that the proposed DCNN architecture was able to provide an accuracy of 89% to 93.67%. SGD optimizer gave the highest accuracy, with the CNN model

VGG16 providing more stable and consistent performance throughout the epoch. Overall, it showed that the higher the epoch, the higher accuracy. The proposed DCNN model managed to give the best results in epoch 25 in both Adam and SGD optimizers and managed to produce the second-best result even in some of the epochs where other CNN models seemed to outperformed it. Future work is set to add more samples in the dataset and its influence on the learning. Other established models will be used as well to allow fine-tuning and target for a better result.

REFERENCES

[1] J. E. Grimm and M. Steinhaus, "Characterization of the Major Odorants in Cempedak - Differences to Jackfruit," *Journal of Agricultural and Food Chemistry*, vol. 68, no. 1, pp. 258–266, 2020, doi: 10.1021/acs.jafc.9b06564.

[2] M. M. H. Wang *et al.*, "Origin and diversity of an underutilized fruit tree crop, cempedak (Artocarpus integer, Moraceae)," *American Journal of Botany*, vol. 105, no. 5, pp. 898–914, 2018, doi: 10.1002/ajb2.1094.

[3] J. Balamaze, J. H. Muyonga, and Y. B. Byaruhanga, "Physico-chemical Characteristics of Selected Jackfruit (Artocarpus Heterophyllus Lam) Varieties," *Journal of Food Research*, vol. 8, no. 4, p. 11, 2019, doi: 10.5539/jfr.v8n4p11.

[4] M. Somashekhar, N. Nayeem, and B. Sonnad, "a Review on Family Moraceae(Mulberry) With a Focus on Artocarpus Species," *World Journal Of Pharmacy And Pharmaceutical Sciences*, vol. 2, pp. 2614–2621, 2013, [Online]. Available: www.wjpps.com.

[5] H. Basri, I. Syarif, and S. Sukaridhoto, "Faster R-CNN implementation method for multi-fruit detection using tensorflow platform," *International Electronics Symposium on Knowledge Creation and Intelligent Computing, IES-KCIC 2018 - Proceedings*, pp. 337–340, 2019, doi: 10.1109/KCIC.2018.8628566.

[6] A. Bhargava and A. Bansal, "Automatic Detection and Grading of Multiple Fruits by Machine Learning," *Food Analytical Methods*, vol. 13, no. 3, pp. 751–761, 2020, doi: 10.1007/s12161-019-01690-6.

[7] K. Bresilla, G. D. Perulli, A. Boini, B. Morandi, L. Corelli Grappadelli, and L. Manfrini, "Single-shot convolution neural networks for real-time fruit detection within the tree," *Frontiers in Plant Science*, vol. 10, 2019, doi: 10.3389/fpls.2019.00611.

[8] M. Buzzelli, F. Belotti, and R. Schettini, "Recognition of edible vegetables and fruits for smart home appliances," *IEEE International Conference on Consumer Electronics - Berlin, ICCE-Berlin*, vol. 2018-September, 2018, doi: 10.1109/ICCE-Berlin.2018.8576236.

[9] F. Femling, A. Olsson, and F. Alonso-Fernandez, "Fruit and Vegetable Identification Using Machine Learning for Retail Applications," *Proceedings - 14th International Conference on Signal Image Technology and Internet Based Systems, SITIS 2018*, pp. 9–15, 2018, doi: 10.1109/SITIS.2018.00013.

[10] S. Jana, S. Basak, and R. Parekh, "Automatic fruit recognition from natural images using color and texture features," *Proceedings of 2nd International Conference on 2017 Devices for Integrated Circuit, DevIC 2017*, pp. 620–624, 2017, doi: 10.1109/DEVIC.2017.8074025.

[11] R. Khan and R. Debnath, "Multi Class Fruit Classification Using Efficient Object Detection and Recognition Techniques," *International Journal of Image, Graphics and Signal Processing*, vol. 11, no. 8, pp. 1–18, 2019, doi: 10.5815/ijigsp.2019.08.01.

[12] N. Lamb and M. C. Chuah, "A Strawberry Detection System Using Convolutional Neural Networks," *Proceedings - 2018 IEEE International Conference on Big Data, Big Data 2018*, pp. 2515–2520, 2019, doi: 10.1109/BigData.2018.8622466.

[13] K. Hameed, D. Chai, and A. Rassau, "A comprehensive review of fruit and vegetable classification techniques," *Image and Vision Computing*, vol. 80, pp. 24–44, 2018, doi: 10.1016/j.imavis.2018.09.016.

[14] M. Rahnemoonfar and C. Sheppard, "Deep count: Fruit counting based on deep simulated learning," *Sensors (Switzerland)*, vol. 17, no. 4, 2017, doi: 10.3390/s17040905.

[15] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu, "A survey on deep transfer learning," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11141 LNCS, pp. 270–279, 2018, doi: 10.1007/978-3-030-01424-7_27.

[16] F. Risdin, P. K. Mondal, and K. M. Hassan, "Convolutional neural networks (CNN) for detecting fruit information using machine learning techniques," *IOSR Journal of Computer Engineering*, vol. 22, no. 2, pp. 1–13, 2020, doi: 10.9790/0661-2202010113.

[17] S. H. Miraei Ashtiani, S. Javanmardi, M. Jahanbanifard, A. Martynenko, and F. J. Verbeek, "Detection of mulberry ripeness stages using deep learning models," *IEEE Access*, vol. 9, pp. 100380–100394, 2021, doi: 10.1109/ACCESS.2021.3096550.

[18] V. Venkatesh, N. Yallappa, S. U. Hegde, and S. R. Stalin, "Fine-Tuned MobileNet Classifier for Classification of Strawberry and Cherry Fruit Types," *Journal of Computer Science*, vol. 17, no. 1, pp. 44–54, 2021, doi: 10.3844/jcssp.2021.44.54.

[19] B. Ma, X. Li, Y. Xia, and Y. Zhang, "Autonomous deep learning: A genetic DCNN designer for image classification," *Neurocomputing*, vol. 379, pp. 152–161, 2020, doi: 10.1016/j.neucom.2019.10.007.

[20] S. S. S. Palakodati, V. R. R. Chirra, Y. Dasari, and S. Bulla, "Fresh and rotten fruits classification using CNN and transfer learning," *Revue d'Intelligence Artificielle*, vol. 34, no. 5, pp. 617–622, 2020, doi: 10.18280/ria.340512.

[21] I. Hussain, Q. He, and Z. Chen, "Automatic Fruit Recognition Based on DCNN for Commercial Source Trace System," *International Journal on Computational Science & Applications*, vol. 8, no. 2/3, pp. 01–14, 2018, doi: 10.5121/ijcsa.2018.8301.

[22] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 2015.

[23] M. Shaha and M. Pawar, "Transfer Learning for Image Classification," *Proceedings of the 2nd International Conference on Electronics, Communication and Aerospace Technology, ICECA 2018*, pp. 656–660, 2018, doi: 10.1109/ICECA.2018.8474802.

[24] N. Sharma, V. Jain, and A. Mishra, "An Analysis of Convolutional Neural Networks for Image Classification," *Procedia Computer Science*, vol. 132, pp. 377–384, 2018, doi: 10.1016/j.procs.2018.05.198.

[25] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2016-December, pp. 2818–2826, 2016, doi: 10.1109/CVPR.2016.308.

[26] S. Gupta, "Classify any Object using pre-trained CNN Model," *Towards Data Science*, 2020, [Online]. Available: https://towardsdatascience.com/classify-any-object-using-pre-trained-cnn-model-77437d61e05f .

[27] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, vol. 2017-January, pp. 1800–1807, 2017, doi: 10.1109/CVPR.2017.195.

BIOGRAPHIES OF AUTHORS

Putra Sumari is a Professor at the School of Computer Sciences, Universiti Sains Malaysia. He received MSc in 1996 at Liverpool University and PhD in 2001 at John Moores University, England. He leads the robotic, computer vision, and image processing research group at Universiti Sains Malaysia. His research interests are image/video analysis, deep learning, neural networks, and brain-computer interface. He can be contacted at email: putras@usm.my.

Azleena Mohd Kassim is a senior lecturer at the School of Computer Sciences, Universiti Sains Malaysia (USM). She received her BSc, MSc, and PhD in Computer Science from USM. She works in multiple domains such as knowledge engineering and management, ontology, artificial intelligence, evolutionary computing, and health informatics. She can be contacted at email: azleena.mk@usm.my.

Song-Quan Ong is a postgraduate, Master in Data Science student at the School of Computer Sciences, Universiti Sains Malaysia (USM). He received his BSc in 2007, MSc in Applied Entomology in 2011, and PhD in Medical and Veterinary Entomology in 2014 from USM. His current research interests include deep learning, agriculture and animal disease diagnosis. He can be contacted at email: songquan.ong@student.usm.my.

Gomesh Nair is currently undergoing the Master in Data Science program at the School of Computer Sciences, Universiti Sains Malaysia. Previously he received his BEng, MSc and Ph.D. at Universiti Malaysia Pahang. His current research interests are deep learning, computer vision, and medical diagnosis. He can be contacted at email: gomesh.nair@student.usm.my.

Al Dabbagh Ragheed is a postgraduate Master's student at the School of Computer Sciences, Universiti Sains Malaysia. He received his BSc from the University of Mosul, Iraq. His current research interests include image processing and data analytics. He can be contacted at email: ragheedduraid@student.usm.my.

Nur Farihah Aminuddin is a postgraduate, Master in Data Science student at the School of Computer Sciences, Universiti Sains Malaysia. She received her BSc in Actuarial Science from Universiti Teknologi Malaysia and her MBA from Universiti Kebangsaan Malaysia. Her current research interests include data analytics and object detection. She can be contacted at email: farihahaminuddin@student.usm.my.