Engineering and Applied Science Research
https://www.tci-thaijo.org/index.php/easr/index
Published by the Faculty of Engineering, Khon Kaen University, Thailand

Improving rotten fruit classification accuracy through fusion of multiple pretrained CNN models

Singgih Tulus Makmud, Adrian Natanael Haryanto and Simeon Yuda Prasetyo*

Computer Science Department, School of Computer Science, Bina Nusantara University, Jakarta, Indonesia

Received 17 June 2023
Revised 17 February 2024
Accepted 22 February 2024

Abstract

Agriculture has been the main source of livelihood in Indonesia. The main sector lives in fruit sales, which are important for our healthcare. That is why it is important to keep the fruits fresh into the market. To address this issue, machine learning can help identify rotten fruits with deep learning techniques to differentiate fresh fruit from rotten fruit. Some deep learning techniques will be used in this classification, such as VGG16, MobileNetV2, Xception, ResNet-50, and InceptionV3, which are the common models used in fruit classification in some research. Every model will use fine tuning to increase its performance. The fusion model method will also be used in the classification to improve the performance of the model. Lastly, all of the models will be compared to see which is the best model for classification. In the end, there is a model with the best accuracy, which is MobileNetV2, with 0.99706 accuracy before the fusion model. The result of the fusion model gives the best accuracy, with 0.99926 accuracy. These results showed that the fusion model was best used for fruit classification and can help supermarkets classify rotten and fresh fruits, preventing more issues in the future. Still, this paper did not represent all of the machine learning classification algorithms. Further research is needed to compare the performance of different deep learning approaches in detecting rotten fruit in supermarkets.

Keywords: Rotten fruit classification, CNN, Fusion model, Transfer learning, Fine-tuning

1. Introduction

Fruit is a basic need in our society and is important to our healthcare, mostly to our agriculture. In Indonesia, fruit is the biggest other horticulture business within 53.56% of the market [1]. Therefore, it is important to keep improving the facility for the detection of rotten fruit using machine learning to easily monitor the rotting progression of the fruit. There are many convolutional neural network (CNN) methods that have been used, such as VGG16, MobileNetV2, etc. The comparison of deep learning methods for the categorization of rotten fruit (VGG16, MobileNetV2, Xception, ResNet-50, InceptionV) and the fusion model is necessary to investigate their performance at classifying fruit ripeness because these architectures have been widely used for a variety of computer vision tasks.

The classification of fruit ripeness and freshness has been a frequent study that has been conducted in some research with many methods from deep learning to classical machine learning. Numerous studies have reported the use of deep learning methods to determining the maturity of fruits.

In [2], for example, outlined a method for decreasing human efforts by decreasing manufacturing costs and times by recognizing fruit flaws in the agriculture sector. The suggested model divides the input fruit photos into rotten and fresh fruit categories. The given input fruit images are categorized using Softmax into fresh and rotting fruits. The characteristics of the input fruit photos are then extracted using a convolutional neural network. The proposed model's accuracy is 97.82%. The results demonstrate that the proposed CNN model can accurately classify both rotting and fresh fruits.

The study of rotten fruit classification in [2] also proposed a method, which is transfer learning from a pre-trained model that is VGG16, MobileNet, and Xception to make a proposed model to later compare the two models. A confusion matrix was also utilized to monitor how well each model was performing. Similar to the method proposed by [3], the effectiveness of the improved models was greater than the model created from the RGB picture dataset because they employed the GrabCut technique, the RGB, Laplacian or Gaussian, and Hue Saturation Value with Adaptive Gaussian Thresholding. With an accuracy for validation of 89.9% and a predicted accuracy of 91.34%, this model can effectively identify whether red fruit is fresh or rotten. Convolutional Neural Networks (CNNs) have outperformed other deep learning architectures in image categorization tasks. Because of their effectiveness and computational economy, VGG16 and MobileNet have been frequently employed for image classification.

Model for Classifying Fruit Images Using MobileNetV2 and Deep Transfer Learning, proposed by [4], that replaced the classification layer of MobileNetV2 to create a modified version of MobileNetV2 called TL-MobileNetV2, which reaches a 99% accuracy rate, is 3% more accurate than MobileNetV2 and has a 1% error rate overall. When compared to AlexNet, InceptionV3, VGG16, and ResNet, the accuracy is improved by 8, 11, 6, and 10%, respectively. In addition, the TL-MobileNetV2 result shows 99% results for precision, recall and F1 score.

*Corresponding author.
Email address: simeon.prasetyo@binus.ac.id
doi: 10.14456/easr.2024.30

A similar study around fruit classification by [5], focuses on a variety of image processing methods, such as InceptionV3, which, when applied to the ImageNet dataset, produced more than 78.1% accuracy. In similar studies conducted by [6], created an automatic fruit classification and detection system utilizing ResNet50 and VGG16 deep learning techniques. With a ResNet50 model result of 86% on the FIDS-30 (Fruit Image Data Set) and 99% accuracy on a new dataset.

In the [7] research paper, by fusing a 3-dimensional convolutional neural network with input from motion pictures and a 2-dimensional convolutional neural network with input from 78 OFMT or Optical Flow-guided Motion Template photos, they recommend using a two-stream fusion model to recognize hand motions. The results conclude that the fusion technique has boosted the recognition result. In similar studies [8], focusing on CNN's influence with fine-tuning method applied with the CNN used were VGG16, VGG19, and InceptionV3.

In similar research by [9], using a deep learning algorithm to classifying fruit and vegetables into rotten and fresh category using an improved model of YOLO named YOLOv4 model with the accuracy of 51.2% accuracy on the testing result. Research conducted by [10], proposed an improved semantic segmentation using Enhanced U-net to classify a rotten portion in a fruit's RGB image. The result of the improved version of the deep learning technique was 97.46% on the training result and 97.54% on validation result. In [11], aims to help several countries that have some limitation to access the technology to classify the ripeness of the fruit using various machine learning approaches adding some system to do some analysis in pre- and post- harvest.

In published research by [12], uses ResNet50 model to classify fruit using the "Fruit 360" dataset with the result of 100% accuracy that resulting in an overfitting of the model that have been made from scratch. The research made by [13], proposed a CNN model to classify fresh and damaged in order to reduce food waste with an accuracy of 97.14 % accuracy which is better than another proposed model by other author. In study paper made by [14], the author constructed a device with CNN model to classify fresh and rotten fruit with the validation accuracy of 98.21% using only 10 epochs. Research by [15], proposed a CNN model to classifying fruits using transfer learning technique which give an accuracy of 98.23%, resulting in a more efficient model than other transfer learning model. Using fusion method, author of [16] proposed a fusion of U-Net model and CNN model to classify skin lesion from demoscopy images with the accuracy of 97.26% with 20 epocs and 32 batch size.

In this paper, VGG16, MobileNetV2, Xception, ResNet-50, and InceptionV3 were used as pre-trained models to transfer learning a model that can classify rotten fruit. After that, the results of a trained model that has been fine-tuned and a fusion model of the best three models' accuracy were compared. This research presents an enhanced version of classifiers for the job of rotting fruit classification, shedding light on the performance of several neural network designs for this application.

2. Materials and methodology

2.1 Dataset gathering

The dataset for fresh and rotten fruits was from Kaggle, with a total of 13599 images consisting of 5904 fresh fruit images and 7695 rotten fruit images. In Figure 1, there are three types of fruits: apples, oranges, and bananas. Next, the dataset was divided into two classes: rotten and fresh. What to keep in mind is that the background of the dataset cannot be in grayscale, it needs to be in color.

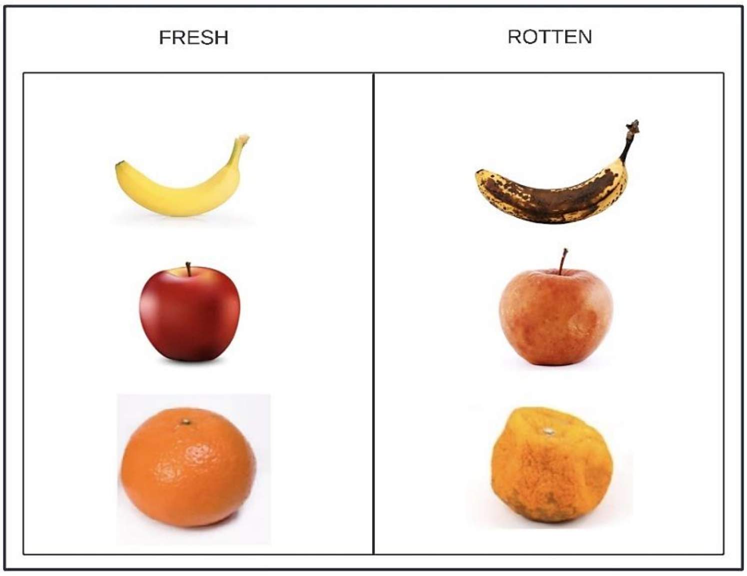

Figure 1 Example of the 2 class datasets

Training, validation, and test set ratio was 80: 10: 10, so the training set consisted of 10881 images, the validation and test sets consisted of 1359 each that belonged to 2 classes, rotten and fresh.

2.2 Pre-Processing data

The dataset originally was divided into 2 files, training and test sets, each consisting of 6 classes, which were 3 types of fruits and their conditions. In this research, the dataset needed to be divided into two classes: fresh and rotten. In that case, some slight adjustments had to be made to the data so that the images were grouped based on their condition, fresh or rotten. Then, using ImageDataGenerator, batches of image data were generated and also normalized or rescaled so that the image's pixel values varied from 0 to 1. The datasets were also resized to 224 x 224.

2.3 Proposed methodology

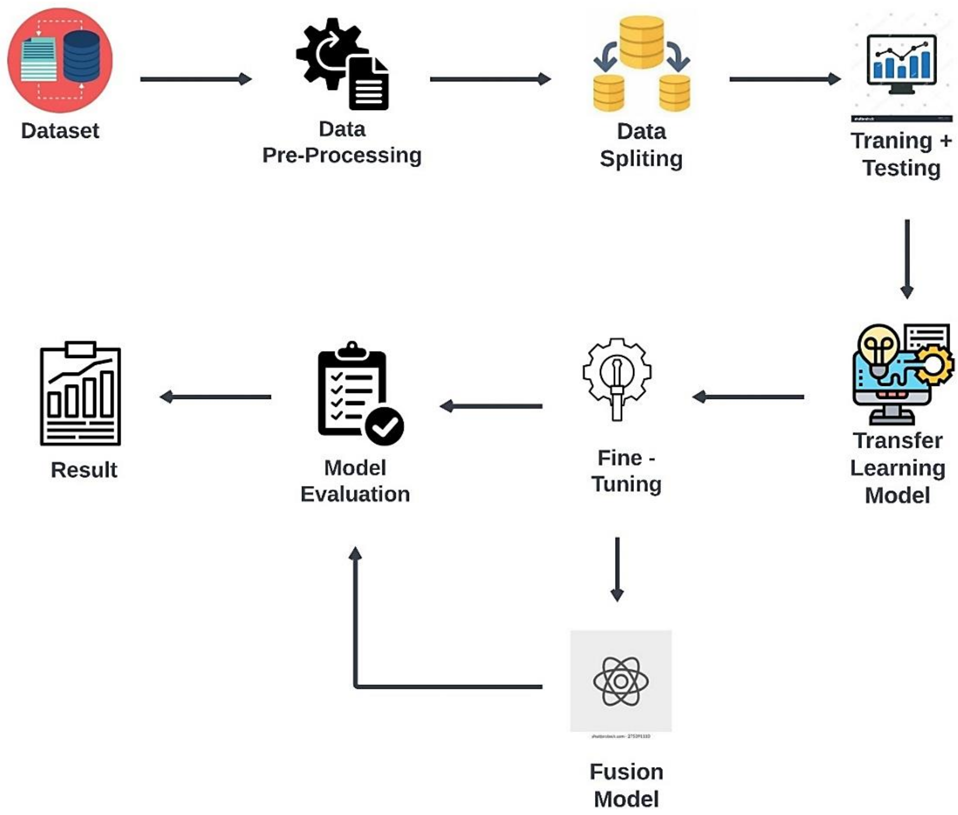

Figure 2 Rotten Fruit Classification workflow.

The dataset from Kaggle was split into two folders, the train and test folders, with 6 classes in each folder. In this classification, 2 classes and 3 (train, validation, test) splits were made. Therefore, all datasets were merged into the 'temp' folder with 2 folders (rotten and fresh). After the folders were made, by using split-folders, using an 80: 10: 10 split, the data were divided into training, validation, and test sets, resulting in a training set consisting of 10881 images, the validation and test sets consist of 1359 images each that belong to 2 classes. Each model was trained for up to 10 epochs to ensure uniformity and for comparison purposes. Figure 2 shows the paper's workflow.

2.4 Convolutional Neural Networks

Convolutional Neural Networks or CNN is a deep learning algorithm that is commonly used to process image and video inputs and can successfully detect or classify distinct objects by their patterns. CNN has many layers; the most notable ones are the convolution layer. It conducts a dot product between a kernel and the constrained region of the receptive field [3]. This layer applies filters to the image input, which later produces a set of feature maps. With widely used procedures like maximum pooling and average pooling, the pooling layer decreases the size of the feature maps. Another layer that is the layer that is entirely linked, which takes the flattened output from the layer and then produces a set of output values.

2.5 Transfer learning

Transfer learning is the process of applying a model that has already been trained to address one problem to another [4]. A source model is needed to do transfer learning, using a pre trained model approach, the selected models are then reused as the starting point. All parts of the model can be used or tuned, depending on the needs.

2.5.1 VGG16

VGG16 is a CNN that is 16 layers deep, based on the name. This model uses very small convolution filters (3 x 3). VGG16 is commonly used for object classification and detection that can classify 1000 images into 1000 different categories [2]. This model takes the input tensor size as (244, 244, 3). The number of classes needs to be specified, in this case, there are 2 classes, rotten and fresh.

2.5.2 ResNet–50

ResNet-50, also called Residual Network, is a deep neural network that, instead of learning unreferenced functions, it picks up residual functions based on the layer's inputs [17]. ResNets create networks by stacking leftover bricks on top of one another. In this case, ResNets uses 50 layers to enhance the network's depth and provide the feature in the classification page [18].

2.5.3 MobileNetV2

MobileNetV2 is also a pre-trained model that is used for detection and classification, but this model has a special feature, which has a very small size so that it can be used on mobile devices, hence the name MobileNetV2 [2]. This model has a layer of convolution containing 32 filters, 19 residual congestion layers, and it is based on an inverted residual structure. This model will also accept image inputs with (244, 244, 3) specifications and change the number of classes to 2, rotten and fresh.

2.5.4 InceptionV3

InceptionV3 is part of the Inception family with a remarkable structural design [6]. A convolutional filter with many variable sizes was suggested in InceptionV3 as an inception model [19]. The InceptionV3 module has 42 layers and employs batch normalization (BN) in the fully connected and convolutional layers. There is a sequence of three convolutional layers at the start of InceptionV3 which accepts an image with dimensions of 299 x 299 x 3 [5].

2.5.5 Xception

Xception, or can be called Extreme Inception, is CNN architecture that was proposed by Chollet [20] from Google. Xception's feature extraction base network is composed of 36 convolutional layers [21]. Xception has an architecture It is a depth-wise separable linear stack of convolution layers with residual connections, making it exceedingly flexible [20]. Xception has the same parameter count as InceptionV3. The result of Xception slightly outperformed InceptionV3 with ImageNet as the dataset and significantly outperformed InceptionV3 on the JFT dataset.

2.6 Fine tuning

A deep learning technique called "fine-tuning" adapts previously trained networks to a new job by "fine-tuning" them, allowing the weights' data to be used for the new task. A pre-trained convolutional neural network's performance may be greatly increased while using less target-labeled data by fine-tuning it on a target dataset [22]. Fine-tuning works best in large dataset by fine – tuning each block-wise to see the effect of each block of the model network [8]. Fine-tuning can make an adjustment to improve the performance of the model [23]. Fine-tuning can be applied in parameter tuning to see the most efficient parameter [24].

2.7 Fusion model

Fusion Model is an approach that, instead of employing a single data model, mixes information from other models to get complementary and comprehensive data that will help machine learning models perform better [25]. The fusion model combines all the prediction of all the model then output the final prediction [26]. Fusion can also be used to reduce system error by fusing the data [27]. The strategy used is Late Fusion Strategy.

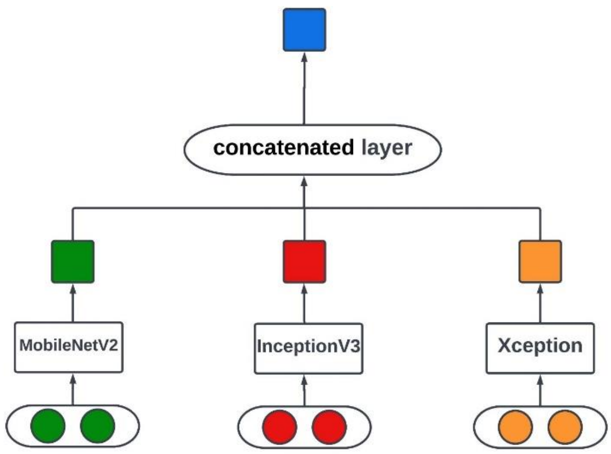

Figure 3 Fusion model visualization.

The process of using predictions from several models to get a definitive conclusion is known as late fusion, also known as decision-level fusion [22]. Different models are trained using various features, and the final choice is made by leveraging multiple sources of information, the overall accuracy of the prediction is improved. The output of each model is first processed independently to produce predictions or features for each data point. The final forecast is then created by separately combining these traits or predictions. The use of the late fusion method is to combining the information from the 3 model which is MobileNetV2, InceptionV3, Xception, so the accuracy of the fusion model can be improved than the accuracy provided by the three model individually. Figure 3 shows the fusion model visualization for the paper.

2.8 Performance metrics

The models are trained and tested on a test set to evaluate whether each model has successfully classified the rotten and fresh fruits or not. The evaluation needs metrics as a value to compare another model. The performance of three models is going to be evaluated using precision, recall, accuracy, and F1-score [28].

$$Accuracy = \frac{TN+TP}{TN+FP+TP+FN} \quad (1)$$

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

$$F1\ Score = 2 * \frac{Precision*Recall}{Precision+Recall+FP} \quad (4)$$

Each metrics has different calculations and implementations. Accuracy (1) is a metric that measures the correctness of the classifications, whereas precision (2) is calculated by obtaining the ratio of correct positive / true positive among the number of all positive predictions. A measurement called recall (3) determines the proportion of genuine positives to the total of true positives and false negatives. F1 score (4) is a metric that represent harmonic mean between sensitivity and precision metric, ranging from 0 to 1 [29]. The use of the 4-evaluation metric is used to determine the correct evaluation for a classification problem.

3. Results and discussion

The code for this paper was developed in Kaggle notebook. Kaggle Notebooks run in a remote computational environment with an accelerator of a single NVIDIA Tesla-P100 GPU, which allows up to 13 gigabytes of RAM and 2 CPU cores. CPU specifications of 20 gigabytes of automatically saved storage space, 30 gigabytes of RAM, and 4 CPU cores. Figure 4 and 5 show the learning curve of MobileNetV2, which is the best transfer learning model after fine-tuning. Figures 6 and 7 depict the accuracy and loss of the fusion model's learning curve. The models' final results are displayed in Table 1.

Table 1 Combined results of all the models used.

Technique	Model	Accuracy	Precision	Recall	F1 Score
Transfer Learning	VGG16	0.91844	0.91738	0.92458	0.91798
	MobileNetV2	0.98898	0.99019	0.98751	0.98875
	Xception	0.99118	0.99122	0.99083	0.99102
	ResNet50	0.60691	0.65386	0.55485	0.49620
	InceptionV3	0.98384	0.98248	0.98493	0.98359
Fine - Tuning	VGG16	0.99559	0.99571	0.99531	0.99551
	MobileNetV2	0.99706	0.99720	0.99681	0.997
	Xception	0.99559	0.99571	0.99531	0.99551
	ResNet50	0.67523	0.70002	0.69331	0.67454
	InceptionV3	0.99706	0.99681	0.99721	0.99701
	Fusion model	0.99926	0.99935	0.99915	0.99925

After the models were trained using the datasets, to further improve the models' quality or accuracy, the models were fine-tuned. Using the same approach, which is dropping the learning rate to 1e-5, the models were tuned. However, the fine-tuned layers on each model differ, adjusting their number of layers. On VGG16, the last block of layers was trained, while the other layers were frozen. This outcome was obtained by freezing all the layers of the model by making it untrainable, and then unfreeze part of the model or layers that is needed to be trained. In this case, the last block of VGG16 consists of 3 convolution layers and 1 pooling layer, from the 15th layer until 18th layer. In the other hand, the last 2 blocks of layers were frozen on MobileNetV2. There were slight differences between each model adjusting to the number of layers of each model. Using the same steps, the 134th layer until 153rd layer were unfrozen. ResNet50 has many layers, resulting in a fine tuning from 143rd layer until 174th layer.

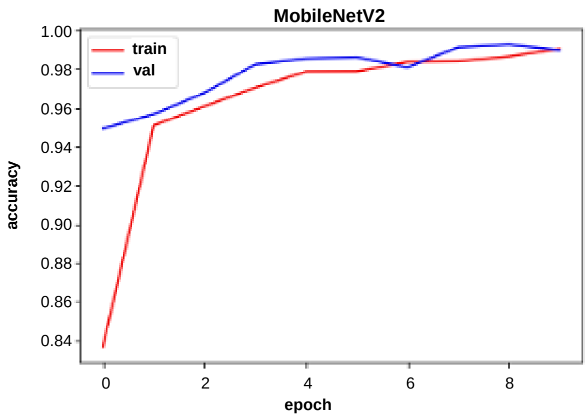

Figure 4 MobileNetV2 before fine-tuning accuracy plot.

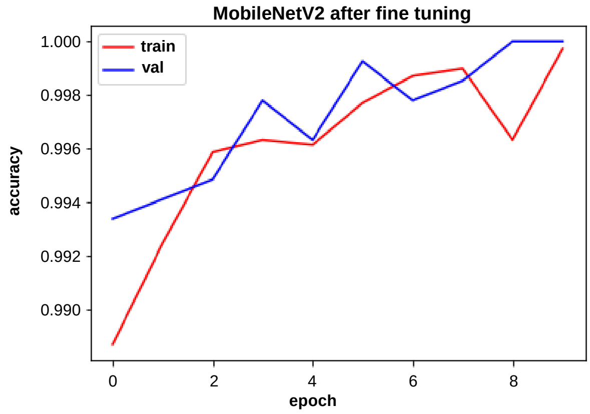

Figure 5 MobileNetV2 after fine-tuning accuracy plot.

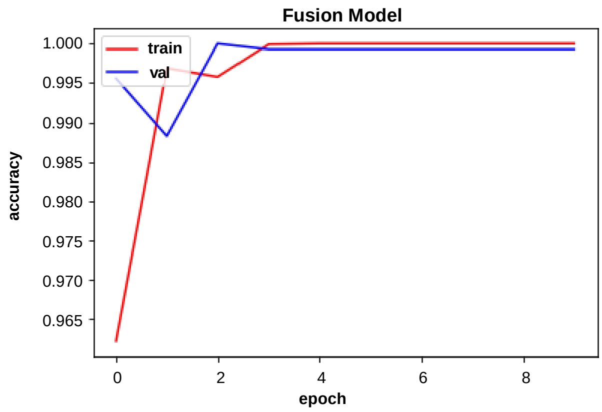

Figure 6 Fusion Model after fine-tuning accuracy plot.

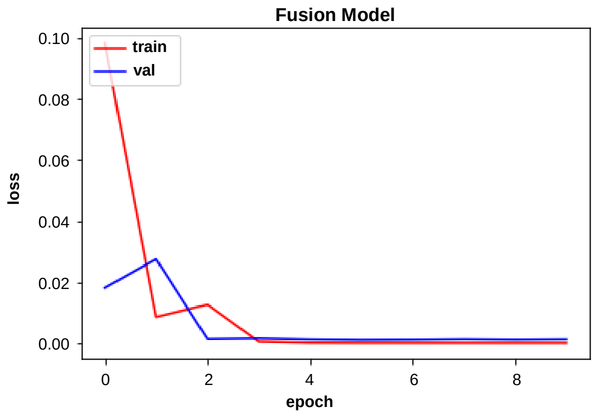

Figure 7 Fusion Model after fine-tuning loss plot.

Table 1 consists of the results of all models used, and it revealed that the fine-tuning method made a slight improvement to its accuracy. Furthermore, the fusion model showed the best results with an accuracy of 0.99926, followed by a precision of 0.99935, recall score of 0.99915, and f1 score of 0.99925 compared to (16) which have only 97.96% accuracy score. Fine tuning a model improved performance metrics because it enables the model to focus or train more on a specific task, in this case by adjusting the trainable or untrainable layers added using Keras: GlobalAveragePooling2D, Dropout(0.2), Dense(256), Dropout(0.2), Dense(256), Dropout(0.2), and Dense(1) with a sigmoid activation because it is a binary classification by freezing it. By doing this, the model could potentially learn more relevant features that could distinguish rotten and fresh fruit while avoiding overfitting.

Each model has a different number of layers or blocks, resulting in a different approach to each model. On the pre-trained model using VGG16, fine tuning starts at the 15th layer, or the last block, so the layer before it was frozen. On MobileNetV2, the last 2 blocks were fine-tuned, starting from the 134th layer. On Xception, the last 2 blocks were fine-tuned, starting from the 116th layer. On ResNet50, the last convolutional block was fine-tuned, starting from the 143rd layer. On InceptionV3, the fine tuning starts at the 258th layer. Those models were compiled while changing the learning rate to 1e-5. All this fine tuning were using different layer instead of the layer used in [8].

In Figure 5, it depicts learning curve of the most best transfer learning model after fine-tuning with the accuracy of 0.99706 compared to before the fine-tuning as shown in Figure 4 with an accuracy of 0.98898. It' shows athesame accuracy with [30] with the accuracy of 99% using the same fine tuning method.

It is further enhanced by using the fusion model method, which combines 3 models with the best performance metrics: Xception, MobileNetV2, and InceptionV3 as shown in Table 1 which is better than [2] which the proposed method shows an accuracy of 0.99926. A concatenated layer was made by combining the output of their pooling layers. The reason why the fusion model was made is because it may improve the accuracy of a model by combining models, as shown in Figure 6 and Figure 7. Each model could fill each other's weaknesses so that a more accurate model could be made.

The result with the use of fusion model combined with fine-tuning method shown was great and could be further improved by more trial and error, such as trying different hyperparameters, different models, different numbers of layers that are fine-tuned, etc. However, based on the current performance metrics, the model could already be tested or implemented in a real-life case, which is rotten fruit classification.

4. Conclusions

In conclusion, in the fruit business, it is important to keep the fruit fresh of the market. To address this issue, the detection of rotten fruit should be improved. CNN, a deep learning method, has demonstrated promise for categorizing fruit ripeness and freshness. Various deep learning architectures have been employed, such as ResNet-50, MobileNetV2, Xception, and VGG16 and InceptionV3. The use of the fusion model has the best accuracy with 99,92% accuracy compared to VGG16, MobileNetV2, Xception, ResNet-50, and InceptionV3 before fine-tuning nor after fine-tuning, with the performance metrics which is : accuracy, precision, recall, and F1-score as the performance comparisons. Fine-tuning makes the accuracy of each model improve, but the fusion model with the fine-tuned model has the best accuracy in the comparison. Additional study is required to compare the performance of different deep learning approaches in detecting rotten fruit in supermarkets and to apply data fusion techniques to agricultural industries to enhance the detection of rotten fruit.

5. References

[1] BPS-Statistics Indonesia. Statistics of horticulture establishments and other horticulture business. Indonesia: Statistics Indonesia; 2022. (In Indonesia)

[2] Palakodati SSS, Chirra VRR, Dasari Y, Bulla S. Fresh and rotten fruits classification using CNN and transfer learning. Revue d'Intelligence Artificielle. 2020;34(5):617-22.

[3] Nuanmeesri S, Poomhiran L, Ploydanai K. Improving the prediction of rotten fruit using convolutional neural network. Int J Eng Trends Technol. 2021;69(7):51-5.

[4] Gulzar Y. Fruit image classification model based on MobileNetV2 with deep transfer learning technique. Sustainability. 2023;15(3):1906.

[5] Khatun M, Ali F, Turzo NA, Nine J, Sarker P. Fruit classification using convolutional neural network (CNN). GRD J Eng. 2020;5(8):1-6.

[6] Mimma NEA, Ahmed S, Rahman T, Khan R. Fruits classification and detection application using deep learning. Sci Program. 2022;2022:1-16.

[7] Sarma D, Kavyasree V, Bhuyan MK. Two-stream fusion model for dynamic hand gesture recognition using 3D-CNN and 2D-CNN optical flow guided motion template. arXiv:2007.08847. 2020:1-7.

[8] Kandel I, Castelli M. How deeply to fine-tune a convolutional neural network: a case study using a histopathology dataset. Appl Sci. 2020;10(10):3359.

[9] Mukhiddinov M, Muminov A, Cho J. Improved classification approach for fruits and vegetables freshness based on deep learning. Sensors. 2022;22(21):8192.

[10] Roy K, Chaudhuri SS, Pramanik S. Deep learning based real-time Industrial framework for rotten and fresh fruit detection using semantic segmentation. Microsyst Technol. 2021;27(9):3365-75.

[11] Africa ADM, Tabalan ARV, Tan MAA. Ripe fruit detection and classification using machine learning. Int J Emerg Trends Eng Res. 2020;8(5):1845-9.

[12] Ukwuoma CC, Zhiguang Q, Bin Heyat MB, Ali L, Almaspoor Z, Monday HN. Recent advancements in fruit detection and classification using deep learning techniques. Math Probl Eng. 2022;2022:1-29.

[13] Kumar TB, Prashar D, Vaidya G, Kumar V, Kumar SD, Sammy F. A novel model to detect and classify fresh and damaged fruits to reduce food waste using a deep learning technique. J Food Qual. 2022;2022:1-8.

[14] Hasan M, Hasan M. Fresh and rotten fruit classification using deep learning. Bangladesh: Daffodil International University; 2021.

[15] Pathak R, Makwana H. Classification of fruits using convolutional neural network and transfer learning models. J Manag Inf Decis Sci. 2021;24(3):1-12.

[16] Anand V, Gupta S, Koundal D, Singh K. Fusion of U-Net and CNN model for segmentation and classification of skin lesion from dermoscopy images. Expert Syst Appl. 2023;213:119230.

[17] He K, Zhang X, Ren S, Sun J. Deep Residual Learning for Image Recognition. arXiv:1512.03385. 2015:1-12.

[18] Mo N, Yan L, Zhu R, Xie H. Class-Specific anchor based and context-guided multi-class object detection in high resolution remote sensing imagery with a convolutional neural network. Remote Sens. 2019;11(3):272.

[19] Nguyen LD, Lin D, Lin Z, Cao J. Deep CNNs for microscopic image classification by exploiting transfer learning and feature concatenation. 2018 IEEE International Symposium on Circuits and Systems (ISCAS); 2018 May 27-30; Florence, Italy. USA: IEEE; 2018. p. 1-5.

[20] Chollet F. Xception: Deep learning with depthwise separable convolutions. arXiv:1610.02357. 2016:1-8.

[21] Gülmez B. A novel deep neural network model based Xception and genetic algorithm for detection of COVID-19 from X-ray images. Ann Oper Res. 2023;328:617-41.

[22] Guo Y, Shi H, Kumar A, Grauman K, Rosing T, Feris R. SpotTune: Transfer learning through adaptive fine-tuning. arXiv:1811.08737. 2018:1-10.

[23] Mustafid A, Pamuji MM, Helmiyah S. A comparative study of transfer learning and fine-tuning method on deep learning models for wayang dataset classification. Int J Inf Dev. 2020;9(2):100-10.

[24] Fu Z, Yang H, So AMC, Lam W, Bing L, Collier N. On the effectiveness of parameter-efficient fine-tuning. arXiv:2211.15583. 2022:1-26.

[25] Huang SC, Pareek A, Seyyedi S, Banerjee I, Lungren MP. Fusion of medical imaging and electronic health records using deep learning: a systematic review and implementation guidelines. NPJ Digit Med. 2020;3(1):136.

[26] Zhen M, Yi M, Luo T, Wang F, Yang K, Ma X, et al. Application of a fusion model based on machine learning in visibility prediction. Remote Sens. 2023;15(5):1450.

[27] Yildiz B, Durdu A, Kayabasi A, Duramaz M. CNN based sensor fusion method for real-time autonomous robotics systems. Turk J Elec Eng Comp Sci. 2022:30:79-93.

[28] Chakraborty S, Shamrat FMJM, Billah MM, Jubair MA, Alauddin M, Ranjan R. Implementation of deep learning methods to identify rotten fruits. 5th International Conference on Trends in Electronics and Informatics (ICOEI); 2021 Jun 3-5; Tirunelveli, India. USA: IEEE; 2021. p. 1207-12.

[29] Mienye ID, Sun Y, Wang Z. An improved ensemble learning approach for the prediction of heart disease risk. Inform Med Unlocked. 2020;20:100402.

[30] Amin U, Shahzad MI, Shahzad A, Shahzad M, Khan U, Mahmood Z. Automatic fruits freshness classification using CNN and transfer learning. Appl Sci. 2023;13914):8087.