

Transfer Learning with Xception Architecture for Snakefruit Quality Classification

Rismaniyati¹, Ardytha Luthfiarta^{*2}

¹Department of Informatics, Faculty of Science and Mathematics, Universitas Diponegoro

²Department of Informatics Engineering, Faculty of Computer Science, Universitas Dian Nuswantoro

E-mail : rismaniyati@live.undip.ac.id¹, ardytha.luthfiarta@dsn.dinus.ac.id^{*2}

Abstract - Machine learning has been greatly used in the field of image classification. Several machine learning techniques perform very well in this task. The development of machine learning technique in recent years are in the direction of deep learning. One of the main challenge of deep learning is that it requires the number of the samples to be extremely large for the model to perform well. This is because the number of feature that trainable parameter are huge. One of the solution to overcome this is by introducing transfer learning. One of the architecture that is currently introduced is Xception architecture. This architecture is claimed to outperform VGG16, ResNet50, and inception in terms of model accuracy and model size. This research aims to classify snakefruit quality by using transfer learning with Xception architecture. This is to explore possibility to achieve better result as Xception architecture generally perform better than other available architecture in transfer learning. The snakefruit quality is classified into two classes. Hyperparameter value is optimized by several scenario to determine the best model. The best performance is achieved by using learning rate of 0.0005, momentum 0.9 and dropout value of 0 or 0.25. The accuracy achieved is 94.44%.

Keywords –transfer learning, Xception, Snakefruit, Deep learning, fruit grading

1. INTRODUCTION

Machine learning has been greatly used in the field of image classification. Several machine learning techniques perform very well in this task. Some of the machine learning algorithm that perform well are Support Vector Machine (SVM) and Artificial Neural Network (ANN). Example of application of classification by using those algorithm is in the field of fruit grading. Research to classify dragon fruit quality was performed by using ANN, SVM and Convolutional Neural Network (CNN). The research is performed based on the shape, size, weight, color, and diseases of fruits[1]. Similar research is performed to classify mulberry fruits. Mulberry is classified into three classes depends on the ripeness of the fruit. The classification is performed based on the the geometrical properties, color, and texture characteristics of the fruit which is classified by using SVM and ANN. Experiment results shows that ANN achieve highest accuracy of 100%[2].

The development of machine learning technique in recent years are in the direction of deep learning. Deep learning enable the model to automatically learn the feature needed for classification by adding convolutional layer before classifier[3]. One of the research on fruit

grading by using deep learning is performed to automatically classify date quality. Dates were classified into 4 classes: Khalal, Rutab, Tamar, and defective date. The algorithm used is CNN with VGG16 architecture. The accuracy achieved in this research is 96.98%[4]. Other research was performed to classify tomatoe quality. Deep residual network was used to detect tomatoes defect automatically and is able to achieve 94.6%[5]. Similar research by using deep learning was performed to classify Okra. This research used three different architecture to classify Okra quality: AlexNet, GoogLeNet and ResNet50. The result shows that ResNet50 achieve highest accuracy of 99%[6].

One of the main challenge of deep learning is that it requires the number of the samples to be extremely large for the model to perform well. This is because the number of feature that trainable parameter are huge. One of the solution to overcome this is by introducing new training paradigm called transfer learning. Transfer learning itself is actually a deep learning model that is applied with pre-trained weight[7]. The transfer learning can be performed by using one of deep learning architecture, such as VGG16, ResNet50, Inception and other available architecture. One of the architecture that is currently introduced is Xception architecture. This architecture is claimed to outperform VGG16, ResNet50, and inception in terms of model accuracy. In addition the model size is also smaller[8].

Example of Previous research on fruit using transfer learning were performed on papaya, tomatoes and Snakefruit[9][10][11] which mostly used VGG16 and ResNet50 to classify fruit quality. Snakefruit itself is local fruit from Indonesia. Other than by using transfer learning, several other researches perform on snakefruit quality classification were performed by using ELM, SVM and CNN[12], [13]. The highest accuracy so far is achieved by VGG16 architecture with 95% accuracy.

This research aims to classify snakefruit quality by using transfer learning with Xception architecture. This is to explore possibility to achieve better result as Xception architecture generally perform better than other available architecture in transfer learning. In this research snakefruit quality is classified into two classes, good and bad quality. Hyperparameter value is being optimized by several scenario to determine the best model.

2. RESEARCH METHOD

The method used in this research is shown in *Figure 1*. First of all, the image data are divided into training data and testing data. After data splitting process, the data are pre-processed to normalize the size and the pixel value. The preprocessed image is then trained with ImageNet pretrained Xception architecture. The model is then saved and used for testing.

2.1. Data

The data used for this experiment is Salak image that is used for previous research (cite). This image is snakefruit dataset used to differentiate good and bad snakefruit. Good snakefruit is snakefruit eligible for export, and bad snakefruit is the one that not eligible. The number of image used in this experiment is 370 images which consist of 190 image from good snakefruit and 180 image from bad snakefruit. Example of good and bad snakefruit image is shown in *Figure 2*.

2.2. Data Split

Total image data of 370 is then split into 80% training data, and 20% testing data. In this case, the total number of training data is 298 and testing data is 72.

2.3. Preprocessing

Images in training data and testing data are then going through preprocessing step. The preprocessing step used in this research is to change the image size to fit the model, which is 299x299x3. After the image size is changed, the pixel is then normalized to a number between 0 to 1 by using (1).

$$f'(x, y) = \frac{f(x, y)}{255} \quad (1)$$

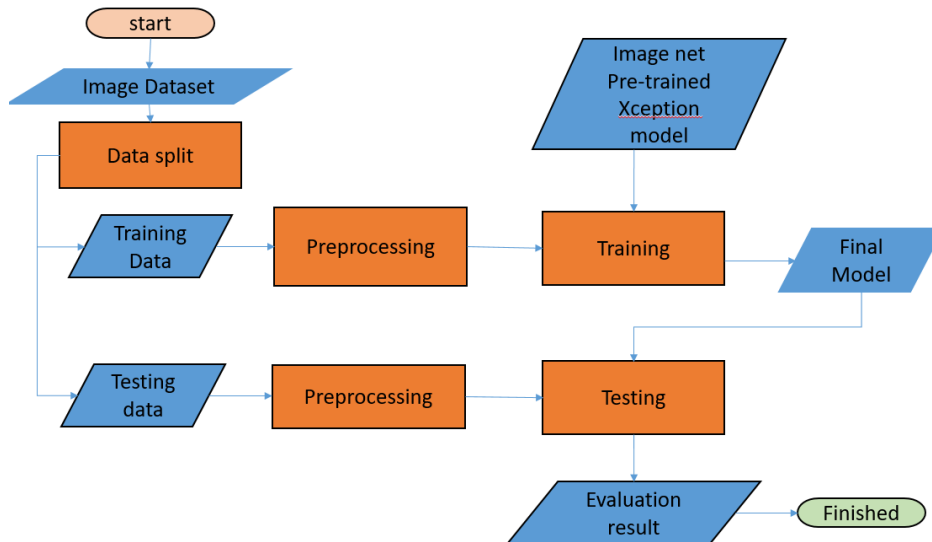


Figure 1. Research Method

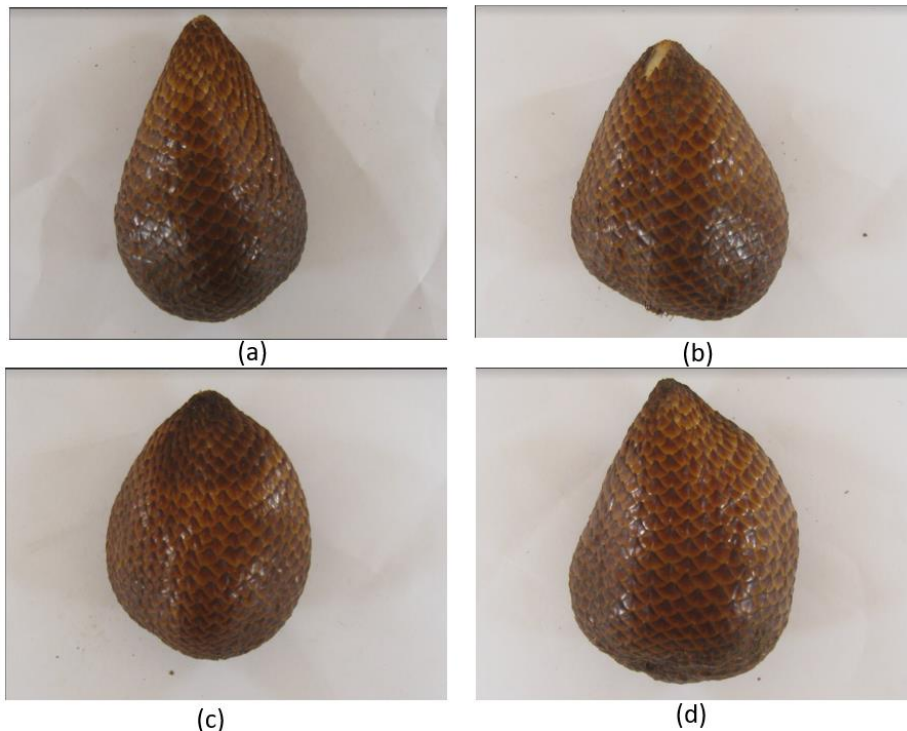


Figure 2. Image example of (a) good snakefruit, (b)-(d) bad snakefruit due to (b) tear, (c) rotten and (d) too ripe.

2.4. Training with ImageNet pre-trained Xception Architecture

After images are preprocessed, the next step is to train by using ImageNet pretrained Xception architecture. ImageNet itself is an image dataset which consists of million of images from one thousand classes. By using Xception architecture that is pretrained in this data set, it is expected that the initial weight used for the training is able to differentiate the image well. The training is then performed to adjust the weight to meet the needs for classifying the snakefruit.

Xception architecture is a deep learning architecture which also known as extreme Inception. This architecture depends on the concept of depthwise separable convolution. The structure of Xception architecture is shown in

Figure 3. It is shown in this figure that Xception architecture input is in the size of 299x299x3. The image is then going through Xception architecture which consist of three main moduls, entry flow, middle flow and exit flow. The entry flow consist of normal convolution followed by depthwise separable convolutions. The middle flow consists of 8 miniblock each consist of 3 depthwise separable convolution. Last but not least the exit flow consists of depthwise separable convolution, feature flattening and output.

The concept of Depthwise separable convolution is to reduce the number of matrix multiplication in the network, hence reducing the number of model size. The structure of depthwise separable convolution is shown in

Figure 4. Instead of performing normal convolution in single step and multiple filter, depthwise separable convolution perform convolution in two steps. The first step is also called depthwise convolution, which uses 1 filter for each input layer. The output of this step will be in the size of *height x width x depth* of the original images, provided padding is used during convolution. The output of depthwise convolution will become the input of pointwise convolution. In pointwise convolution, the size of the filter is 1x1xdepth, where depth is the depth of previous input. To produce multiple feature, more than one filters are used in this process.

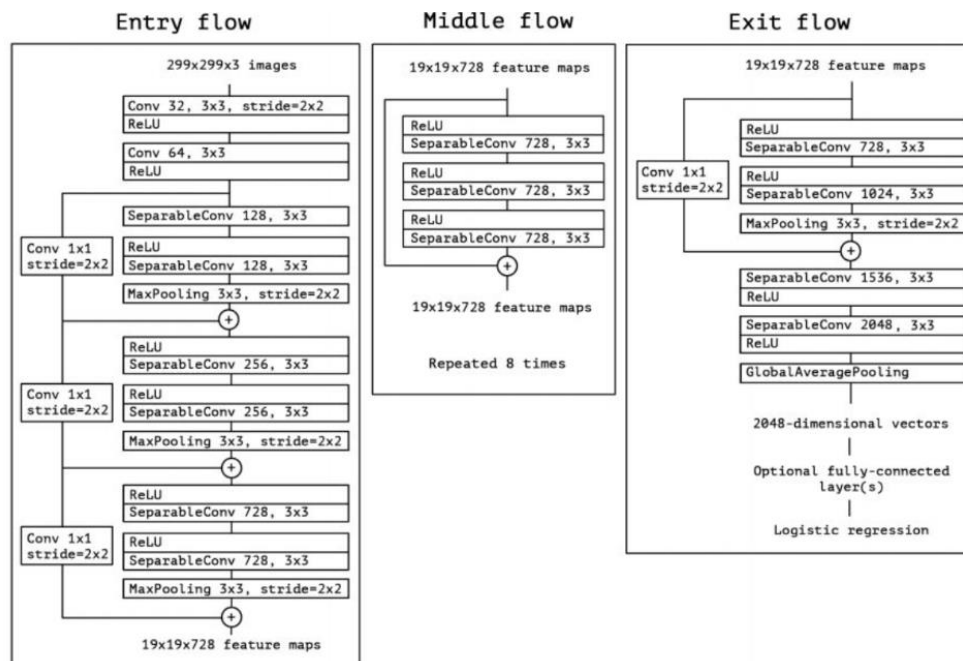


Figure 3. Structure of Xception Architecture[8]

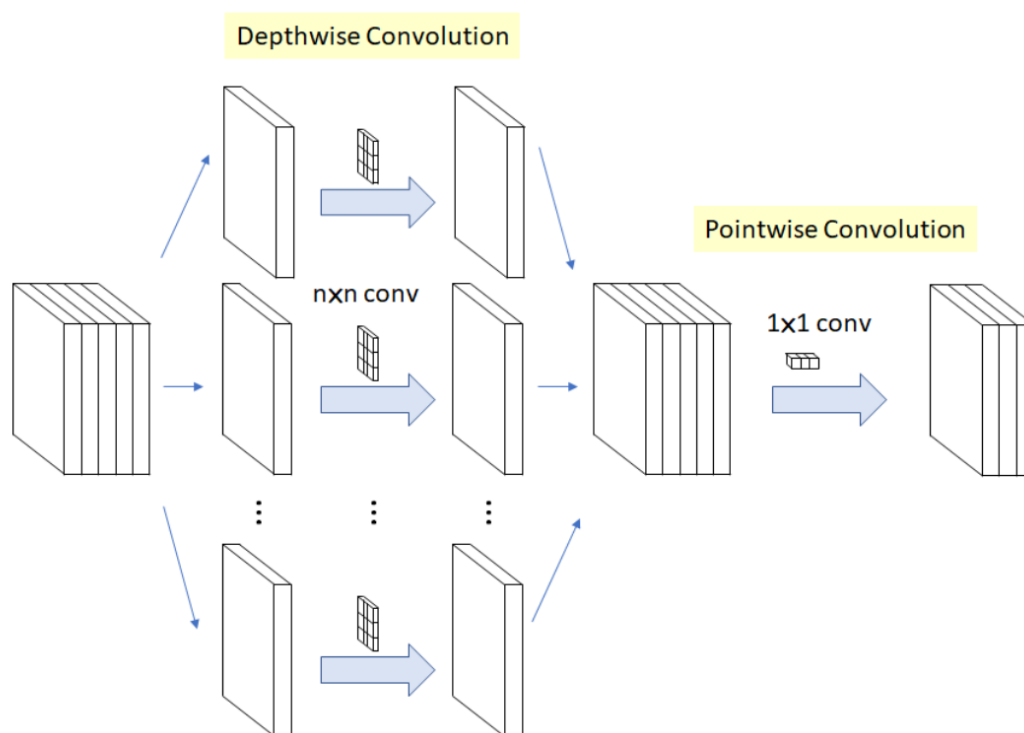


Figure 4. Concept of Depthwise separable convolution

2.5. Model Evaluation

The model produced by the training process will be evaluated. The evaluation will be performed to measure the precision, recall, and accuracy of the model to determine the best model. In order to perform this calculation, model will be applied to validation data. From this process, confusion matrix of the model which consist of true positive, true negative, false positive, and false negative value will be produced. The illustration of confusion matrix is shown in

Figure 5.

Confusion Matrix		Predicted class	
		Positive	Negative
Actual class	Positive	TP	FN
	Negative	FP	TN

Figure 5. Illustration of confusion matrix

True positive (TP) is the number of correctly classified image from good data. True negative (TN) is the number of correctly classified image from the bad data. False positive (FP) is the number bad data that is classified by the models into good class. False negative (FN) is the number good data that is classified by the models into bad class.

From the confusion matrix, the precision, recall and accuracy of the model can be calculated by using (2)-(4). Precision is a measure to understand the extent to which positive prediction is actually correct.

Recall is a measure to calculate how many actual good data is correctly classified. While accuracy is measure which calculate how many total correct prediction as compared to the number of overall data.

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

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

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (4)$$

2.6. Experiment Scenarios

The scenarios used in this experiment are summarized in Table 1. These scenarios are used to find the optimal hyperparameter to produce the best model. All scenarios are performed with 100 epochs.

First scenario is to find the best learning rate. In the first scenario, learning rate is varied from 0.001 to 0.00001. The second scenario is used to find the momentum value to be used in training to improve the accuracy and convergence speed. In the second scenario, the momentum is varied between 0, 0.5 and 0.9. The third scenario is used to find the impact of dropout layer to improve the model performance. In this scenario, the dropout is applied to the best model found in scenario 1 and 2. The dropout value is varied from 0.1 to 0.9 with increment of 0.1.

Table 1. Scenario used in this experiment

Scenario no	Hyperparameter	Condition
1	Learning rate (lr)	Lr = 0.00001, 0.00005, 0.0001, 0.0005, 0.001
2	Momentum	Momentum 0, 0.5, and 0.9
3	Dropout	Without dropout, dropout = 0.25, 0.5, and 0.75

3. RESULTS AND DISCUSSION

The result of the experiment is conducted in three scenarios. First scenario is to find the best learning rate, second scenario is to find the best momentum to be used, and the third scenario is to find the best dropout for the model.

3.1. Result of Scenario 1

Experiment result for scenario is shown in

Figure 6. This figure shows the training accuracy for different learning rate. The learning rate used for this experiment is 0.00001, 0.00005, 0.0001, 0.0005, and 0.001. The experiment result shows that the smaller the learning rate, the slower the training reach the convergence. This is because the smaller coefficient is used to update the weight during the training.

Each model is evaluated by calculating their precision, recall, and accuracy value. The calculation result is summarized in Table 2. The highest precision is 96.43% and is achieved

with learning rate of 0.00005. This model only classify 1 bad snakefruit to good class. However the recall for this model is only 75% with 9 good snakefruit images classified into bad class. The best accuracy and recall of 90.28% and 88.89% is achieved with learning rate of 0.0005. This model classify 3 bad snakefruit images to good class, and 4 good snakefruit images to bad class. Further experiment will be performed by using learning rate of 0.0005.

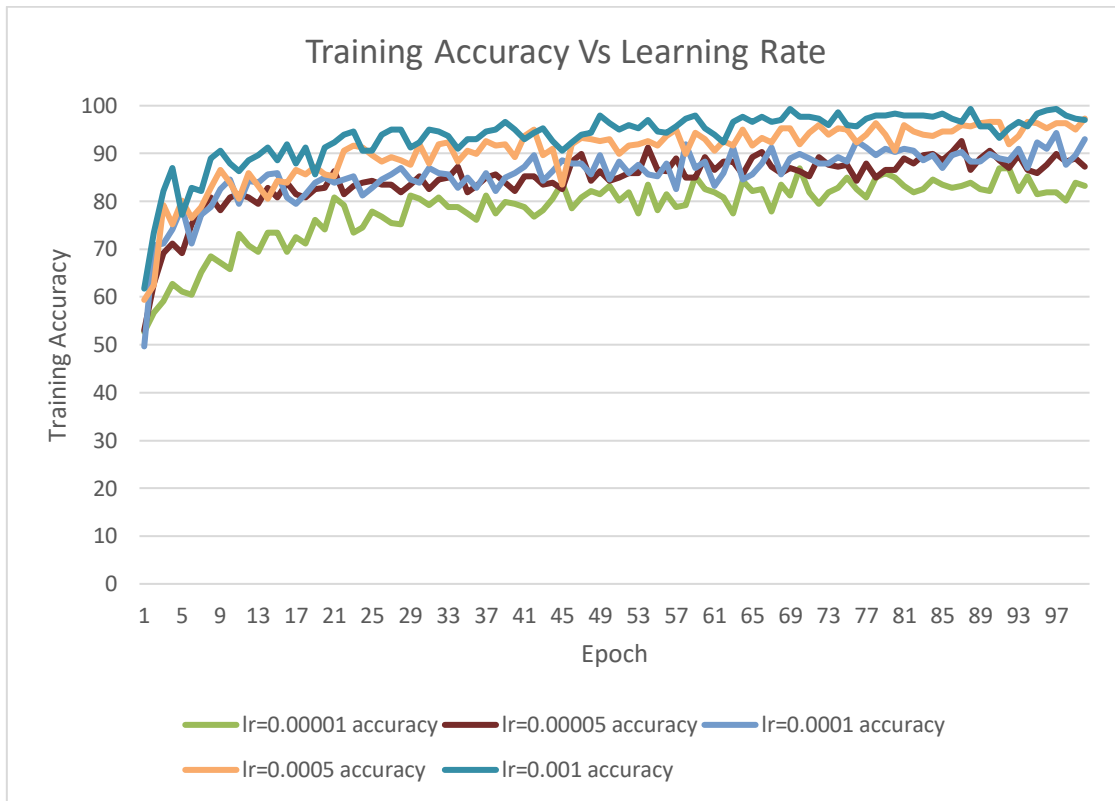


Figure 6. Training accuracy for different Learning rate

Table 2. Precision and Recall for Scenario 1

	TP	TN	FP	FN	Precision	Recall	Accuracy
lr=0.00001	30	30	6	5	83.33%	85.71%	84.51%
lr=0.00005	27	35	1	9	96.43%	75.00%	86.11%
lr=0.0001	30	34	2	6	93.75%	83.33%	88.89%
lr=0.0005	32	33	3	4	91.43%	88.89%	90.28%
lr=0.001	29	33	3	7	90.63%	80.56%	86.11%

3.2. Result of Scenario 2

The second scenario examine the effect of using momentum of 0, 0.5 and 0.9 to improve the model performance. The training process of the second scenario is shown in

Figure 7. It is shown in the first ten epochs of the training that the use of bigger momentum value (0.9), enable the model to converge earlier. This is because the momentum pushes higher weight change during the training.

The model performance for each model is summarized in Table 3. It is shown in this table that the highest precision, recall and accuracy is achieved when the momentum value is 0.9. The precision, recall and accuracy value are 92.11%, 97.77% and 94.44% respectively. This is achieved by classifying 3 bad images to good class and only 1 good images to bad class. From

this result, further experiment in scenario 3 will be performed by using learning rate 0.0005 and momentum of 0.9.

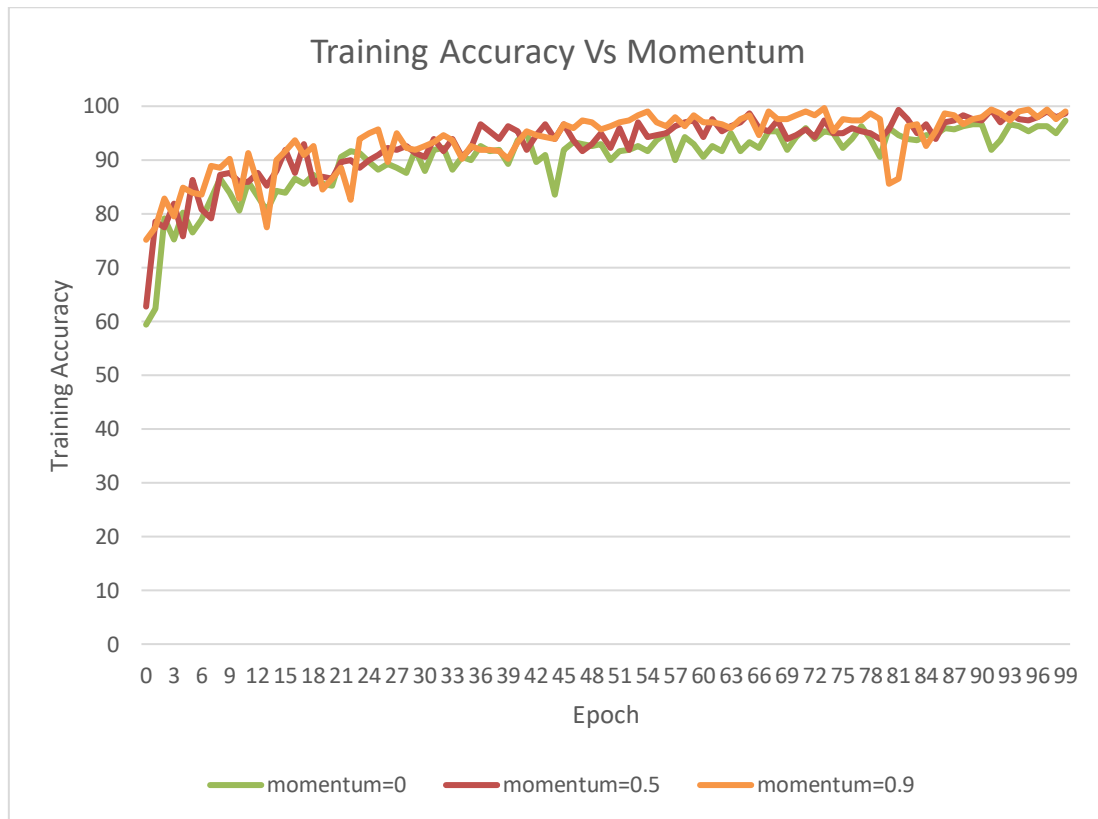


Figure 7. Training accuracy for different momentum

Table 3. Model performance for different momentum

	TP	TN	FP	FN	Precision	Recall	Accuracy
momentum 0	32	33	3	4	91.43%	88.89%	90.28%
momentum 0.5	35	32	4	1	89.74%	97.22%	93.06%
momentum 0.9	35	33	3	1	92.11%	97.22%	94.44%

3.3. Result of Scenario 3

The result of scenario 3 is shown in Table 4. In this scenario, dropout value is introduced to the dense layer or the last layer of the model. The dropout value used is 0, 0.25, 0.5 and 0.75. The experiment result shows that the higher dropout value used will reduce the accuracy. This is because the last layer only consist of 1000 hidden neuron, and reducing too much connection to this number will affect the model accuracy. The best accuracy is achieved with dropout value 0 and 0.25. Both achieve highest accuracy of 94.44%.

Table 4. Model performance for different dropout value

	TP	TN	FP	FN	Precision	Recall	Accuracy
dropout 0	35	33	3	1	92.11%	97.22%	94.44%
dropout 0.25	35	33	3	1	92.11%	97.22%	94.44%
dropout 0.5	29	34	2	7	93.55%	80.56%	87.50%
dropout 0.75	30	34	2	6	93.75%	83.33%	88.89%

4. CONCLUSION

Experiment to evaluate the performance of transfer learning method with Xception architecture for Snakefruit classification has been performed. The experiment is performed in three scenarios. Each scenarios is performed to achieve best hyperparameter for the model. First scenario is performed to determine the best learning rate. The result shows learning rate of 0.0005 achieve best accuracy of 90.28%. The second scenario is performed to determine best momentum to be used. Learning rate value is kept at 0.0005 based on the result of first experiment. The result of second scenario shows that momentum of 0.9 gives best accuracy of 94.44%. The result of second scenario is used to conduct third scenario. The third scenario aims to determine the best dropout value. It is shown in third scenario that dropout value of 0 and 0.25 gives best accuracy. Overall, the best performance is achieved by using learning rate of 0.0005, momentum 0.9 and dropout value of 0 or 0.25. The accuracy achieved is 94.44%.

REFERENCES

- [1] P. U. Patil, S. B. Lande, V. J. Nagalkar, S. B. Nikam, and G. C. Wakchaure, "Grading and sorting technique of dragon fruits using machine learning algorithms," *J. Agric. Food Res.*, vol. 4, no. December 2020, p. 100118, 2021.
- [2] H. Azarmdel, A. Jahanbakhshi, S. S. Mohtasebi, and A. R. Muñoz, "Evaluation of image processing technique as an expert system in mulberry fruit grading based on ripeness level using artificial neural networks (ANNs) and support vector machine (SVM)," *Postharvest Biol. Technol.*, vol. 166, no. December 2019, p. 111201, 2020.
- [3] Ian Goodfellow, Yoshua Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.
- [4] A. Nasiri, A. Taheri-Garavand, and Y. D. Zhang, "Image-based deep learning automated sorting of date fruit," *Postharvest Biol. Technol.*, vol. 153, no. April, pp. 133–141, 2019.
- [5] A. Z. da Costa, H. E. H. Figueroa, and J. A. Fracarolli, "Computer vision based detection of external defects on tomatoes using deep learning," *Biosyst. Eng.*, vol. 190, pp. 131–144, 2020.
- [6] M. M. Raikar, S. M. Meena, C. Kuchanur, S. Girraddi, and P. Benagi, "Classification and Grading of Okra-ladies finger using Deep Learning," *Procedia Comput. Sci.*, vol. 171, no. 2019, pp. 2380–2389, 2020.
- [7] F. Zhuang et al., "A Comprehensive Survey on Transfer Learning," *Proc. IEEE*, vol. 109, no. 1, pp. 43–76, 2021.
- [8] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017*, vol. 2017-January, pp. 1800–1807, 2017.
- [9] S. K. Behera, A. K. Rath, and P. K. Sethy, "Maturity status classification of papaya fruits based on machine learning and transfer learning approach," *Inf. Process. Agric.*, vol. 8, no. 2, pp. 244–250, 2021.
- [10] N. Begum and M. K. Hazarika, "Maturity detection of tomatoes using transfer learning," *Meas. Food*, vol. 7, no. December 2021, p. 100038, 2022.
- [11] R. Rismiyati and A. Luthfiarta, "VGG16 Transfer Learning Architecture for Salak Fruit Quality Classification," *Telematika*, vol. 18, no. 1, p. 37, 2021.
- [12] Rismiyati and H. A. Wibawa, "Snake Fruit Classification by Using Histogram of Oriented Gradient Feature and Extreme Learning Machine," *ICICOS 2019 - 3rd Int. Conf. Informatics Comput. Sci. Accel. Informatics Comput. Res. Smarter Soc. Era Ind. 4.0, Proc.*, pp. 2–6, 2019.

- [13] Rismiyati and S. N. Azhari, "Convolutional Neural Network implementation for image-based Salak sortation," *Proc. - 2016 2nd Int. Conf. Sci. Technol. ICST 2016*, pp. 77–82, 2017.