Bulletin of Electrical Engineering and Informatics
Vol. 14, No. 2, April 2025, pp. 1126~1137
ISSN: 2302-9285, DOI: 10.11591/eei.v14i2.8651

Advancing palm oil fruit ripeness classification using transfer learning in deep neural networks

Rudi Kurniawan[1], Samsuryadi[2], Fatma Susilawati Mohamad[3], Harma Oktafia Lingga Wijaya[4], Budi Santoso[5]

[1]Department of Computer System Engineering, Faculty of Engineering Science, Universitas Bina Insan, Lubuklinggau, Indonesia
[2]Department of Informatics Engineering, Faculty of Computer Science, Universitas Sriwijaya, Palembang, Indonesia
[3]Department of Information Technology, Faculty of Informatics and Computing, University Sultan Zainal Abidin, Kuala Terengganu, Malaysia
[4]Department of Information System, Faculty of Engineering Science, Universitas Bina Insan, Lubuklinggau, Indonesia
[5]Department of Informatics, Faculty of Engineering Science, Universitas Bina Insan, Lubuklinggau, Indonesia

Article Info

Article history:

Received May 1, 2024
Revised Nov 4, 2024
Accepted Nov 19, 2024

Keywords:

Convolutional neural network
Deep learning
Image classification
Palm oil industry
Transfer learning

ABSTRACT

The palm oil industry is a significant component of Indonesia's economy, driven by increasing global demand across various industries. Manual identification of palm oil fruit ripeness is often subjective and labor-intensive, creating a need for a faster and more accurate solution. This study proposes the use of deep learning models based on transfer learning to enhance the classification of palm oil fruit ripeness. Our research evaluates several models, finding that ResNet152V2 achieves the highest performance with superior accuracy and the lowest validation loss. DenseNet201, MobileNet, and InceptionV3 also deliver strong results, each demonstrating an accuracy above 0.99 and a validation loss below 0.04. Cross-validation confirms that ResNet152V2, DenseNet201, and MobileNet maintain high and consistent performance across different folds, showcasing their stability and reliability. This approach provides a promising alternative to manual methods, offering a more efficient and precise means for determining palm oil fruit ripeness, which could significantly benefit the industry by streamlining quality control processes.

This is an open access article under the CC BY-SA license.

Corresponding Author:

Samsuryadi
Department of Informatics Engineering, Faculty of Computer Science, Universitas Sriwijaya
Palembang, Indonesia
Email: samsuryadi@unsri.ac.id

1. INTRODUCTION

The palm oil sector holds a significant role in Indonesia's economy. Since the early 20th century, palm oil has emerged as one of the leading commodities in Indonesia's agricultural sector [1], [2]. The industry is experiencing significant growth due to the increasing global demand for palm oil, which serves as a raw material in various sectors such as food, cosmetics, and biofuels [3]. Traditionally, palm oil fruit identification has been conducted manually by either directly observing the fruit in the field or examining photographs. However, this manual process is often subject to human error and requires considerable time and effort, especially when conducted on a large scale. Deep learning offers a promising alternative by enabling fast and accurate identification of palm oil fruit, which can be integrated into image-based monitoring systems [4]–[9].

Transfer learning is a widely adopted strategy in the development of deep learning models. It offers several advantages for palm oil fruit identification, including: i) the ability to leverage knowledge gained from large datasets to solve new problems, thereby reducing the time and resources needed to train a model from scratch [10]; ii) enhanced performance in image recognition tasks by using pre-trained models that are already adept at recognizing complex features [11]; iii) adaptability to smaller or specific datasets through fine-tuning, allowing the model to adjust existing feature representations to better fit new data [12]; and iv) mitigation of overfitting, particularly when the available dataset is limited, as pre-trained models generally possess robust generalization capabilities [13]. Pre-trained models have usually been trained on very large datasets. Therefore, they have good generalization capabilities. In various applications, transfer learning has proven effective in image recognition tasks. For instance, the VGG19 and DenseNet models have been employed in the health sector for breast cancer detection, addressing the challenge of imbalanceddata [14], [15]. Similarly, the ResNet model has been utilized for real-time underwater object detection [16], the MobileNet model for weld defect detection [17], the Inception model for pulmonary disease detection [18], and the Xception model for malware classification [19]. These examples underscore the utility of transfer learning models in producing efficient, accurate, and adaptable solutions for specific image recognition tasks, including palm oil fruit identification.

Numerous studies have been conducted on the classification of palm oil fruit ripeness, each contributing to the advancement of this field through the application of various techniques and methodologies. For instance, Septiarini *et al.* [20], focuses on the segmentation of palm oil fruits using a contour-based approach, integrated with the canny algorithm and morphological operations. This method successfully achieved an average accuracy of 90.13%, demonstrating the effectiveness of contour-based segmentation for this task. However, despite the promising results, the accuracy could be further enhanced by incorporating more advanced deep-learning models that can automatically learn complex features from the images. Septiarini *et al.* [21], proposed a method that combines color and texture features for feature selection and classification, utilizing principal component analysis (PCA) for dimensionality reduction and an artificial neural network (ANN) for classification. The method achieved an impressive accuracy of 98.3%. While this method is effective, it primarily relies on handcrafted features, which may not capture the full complexity of the fruit's ripeness characteristics. With the advent of deep learning, transfer learning models such as VGGNet, ResNet, and Inception could potentially offer better feature extraction capabilities, leading to improved classification performance.

Alfatni *et al.* [22], also explored the use of color features combined with ANNs for ripeness identification, achieving an accuracy rate of approximately 94%. This approach underscores the importance of color as a key feature in ripeness classification. However, it also highlights the need for robust algorithms that can handle the variability in lighting conditions and fruit appearances, which can significantly impact the accuracy of the model. Transfer learning models, which are pre-trained on large datasets, could offer a solution by providing more generalized features that are less sensitive to such variations. Alfatni *et al.* [23], introduces a real-time classification system using CCD camera sensors and various image processing techniques. This study's use of multiple feature extraction methods, including Gabor waves and grey level co-occurrence matrix (GLCM), in conjunction with supervised classifiers like support vector machines (SVM), K-nearest neighbor (KNN), and ANN, demonstrates a comprehensive approach to ripeness classification. Although ANN outperformed other classifiers, the study indicates that optimizing the model's architecture and hyperparameters could further improve accuracy and processing speed, especially when applied to real-time systems. Here, transfer learning models could play a pivotal role by offering pre-trained networks that require less computational power while maintaining high accuracy.

Other research conducted by Mansour *et al.* [24], object detection algorithms such as MobileNetV2 SSD, EfficientDet, and YOLOv5 were employed to classify the ripeness of palm oil fruits. YOLOv5m, in particular, showed promising results with a high average precision of 0.842. This highlights the effectiveness of advanced object detection models in handling complex classification tasks. However, the study also suggests that further fine-tuning of these models, including data augmentation techniques and hyperparameter optimization, could lead to even better performance. Furthermore, Shiddiq *et al.* [25], combined hyperspectral imaging with ANN to predict palm oil fruit ripeness, achieving a highest accuracy of 90%. While hyperspectral imaging provides rich information about the fruit's chemical composition, the integration with machine learning models like ANN opens up new possibilities for improving classification accuracy. Nevertheless, the challenge lies in the high dimensionality of hyperspectral data, which could be mitigated by employing dimensionality reduction techniques or transfer learning models that are capable of handling such complex data.

From the previous studies, it is evident that there is still significant potential to enhance the accuracy of palm oil fruit ripeness identification. While various algorithms and methods have been explored, including image processing, neural networks, and object detection, the application of transfer learning models represents a promising direction for future research. These models, which are pre-trained on large-scale datasets, have the ability to generalize better across different datasets, potentially leading to improved classification accuracy. Moreover, the use of appropriate data augmentation techniques and the fine-tuning of hyperparameters can further optimize the performance of these models, making them more robust and accurate for real-world applications. In conclusion, although substantial progress has been made in the classification of palm oil fruit ripeness, the integration of advanced transfer learning models with optimized data processing techniques holds the promise of achieving even higher accuracy and reliability, paving the way for more efficient and automated palm oil harvesting processes.

Extensive experiments have demonstrated the superior accuracy and robustness of transfer learning-based models such as VGGNet, ResNet, MobileNet, DenseNet, Inception, and Xception. This research contributes to the field by applying these advanced transfer learning techniques to the specific task of palm oil fruit ripeness classification, showcasing their potential to improve the efficiency and accuracy of this critical agricultural process. By leveraging state-of-the-art models and fine-tuning them for the unique characteristics of palm oil fruit, our study provides a valuable framework for future research and practical applications in the industry. Additionally, our approach highlights the potential for scaling up automated fruit classification systems, which could significantly reduce manual labor and enhance overall productivity in palm oil plantations.

2. METHOD

This study aims to classify images of palm oil fruit into three categories: raw, ripe, and rotten, with a focus on achieving high accuracy. Accurate classification of the ripeness of palm oil fruit is crucial for optimizing the harvesting process and maintaining product quality. The research method is illustrated in the flowchart presented in Figure 1.

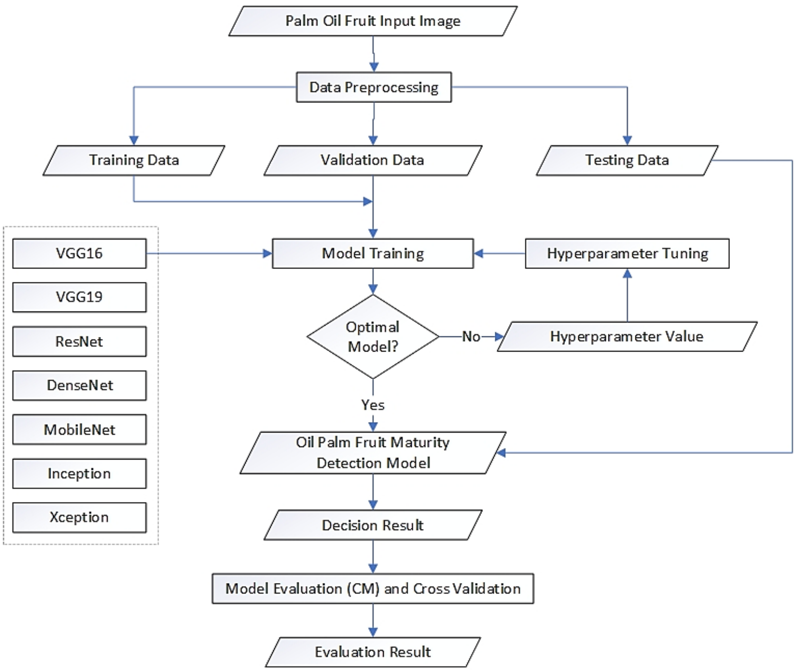


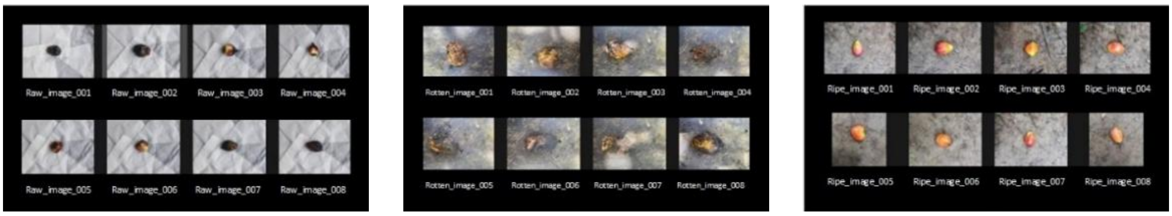
Figure 1. Research method

2.1. Materials

The dataset was collected from community plantations in Durian Remuk village, Muara Beliti district, Musi Rawas regency, South Sumatra, Indonesia. This area spans approximately 394 hectares, with the majority of the population working as palm oil farmers. Images were captured using a 128-megapixel mobile phone camera in JPG format. Samples of each class are illustrated in Figure 2. The dataset consists of 1,500 images, equally distributed across the three categories of palm oil fruit: raw (Figure 2(a)), rotten (Figure 2(b)), and ripe (Figure 2(c)).

2.2. Pre-processing

The raw data, comprising 450 training images and 50 testing images for each class, were organized into two folders, train and test. Validation data consisted of 20% of the training set (1,080 training, 270 validation, and 150 testing images). To facilitate the training process, all images were resized to 640×640 pixels and saved in PNG format. Initially, background removal applications were employed to isolate the foreground by removing the background of the images.



(a) (b) (c)

Figure 2. Palm oil fruit dataset: (a) raw palm oil fruit, (b) rotten palm oil fruit, and (c) ripe palm oil fruit

2.3. Feature extraction (convolutional neural network-based architecture)

This section presents an overview of convolutional neural networks (CNNs) and emphasizes their use in this study through transfer learning. CNNs represent a type of ANN architecture that excels at processing and interpreting image data [26], [27]. Inspired by the way the human brain processes visual input [28], CNNs consist of two primary layers. The first layer is dedicated to feature extraction and feature reduction, and includes three main components: the input layer, the convolution layer, and the pooling layer. Together, these subparts form a feed-forward network, as shown in (1):

$$\theta(X) = \theta_Z\,(\ldots .\theta_2(\theta_1(X, \delta^{(1)}, \delta^{(2)}, \ldots \delta^{(Z)}) \qquad (1)$$

δ, Z, and θ are variables used in this context. The convolution layer analyzes input images to extract important features and generate a feature map (f_m). The kernel is an essential part of the convolution layer and plays a key role in generating the feature map. After the feature map is created, it is forwarded to the next layer, as illustrated in (2):

$$X_t^{(l)} = \theta\,(\textstyle\sum_{i=1}^{M} \omega_{ji}^{(l)} * X_j^{(l-1)} + b_i^{(l)} \qquad (2)$$

where, we have the convolution operator denoted as *, and the i^{th} feature map denoted as $X_j^{(l-1)}$. In order to process each object, a non-linear function (ReLU) is applied. This can be represented as shown in (3):

$$z_{abc} = \max_c(0, x_{abc}) \qquad (3)$$

The activated value for the c^{th} component of the feature map x_{abc} can be located in the convolution layer's output. The final subcomponent of the first layer, known as the pooling layer, is responsible for achieving spatial invariance. This is generally accomplished by down-sampling the feature map (f_m), where adjacent values are combined into a single unit, reducing the input data's dimensionality. Max-pooling is a widely used technique for this purpose. In the network's second layer, the fully connected layer, values from the feature map are fed as inputs. The output of this layer becomes the input for the Softmax layer, which performs classification. The Softmax function is expressed in (4):

$$\gamma\,(Y) = \frac{e^{x_i}}{\sum_{j=1}^{N} e^{x_i}} \qquad (4)$$

Using this approach, the CNN can successfully classify input data. Additionally, numerous studies in the literature have examined CNN performance in detail [29], [30].

2.3.1. VGGNet

CNN architecture introduced in 2014, achieved second place in the ILSVRC 2014 competition [31]. The architecture consists of 16 convolutional layers, three fully connected layers, and five max-pooling layers. Although it is computationally demanding due to its high parameter count, VGGNet has shown exceptional performance in image classification tasks.

2.3.2. ResNet

Developed by Microsoft Research in 2015, is known for its ability to train extremely deep networks effectively [32], [33]. This is achieved through the use of residual connections, which serve as shortcuts to bypass certain layers in the network. These connections help mitigate the vanishing gradient problem, enabling the successful training of much deeper models and enhancing overall network performance.

2.3.3. MobileNet
Designed by Google in 2017, MobileNet is a CNN architecture specifically optimized for mobile devices and resource-constrained environments [34]. It achieves this efficiency by employing depthwise separable convolutions, which significantly reduce the number of parameters and computational requirements. This design enables MobileNet to deliver high performance while maintaining a lightweight structure suitable for real-time applications.

2.3.4. DenseNet
Introduced in 2017 by researchers from Cornell University, DenseNet is a neural network architecture that connects each layer to every other layer within a block [35]. This dense connectivity ensures efficient information flow throughout the network, minimizing redundancy and enhancing gradient propagation. As a result, DenseNet enables the network to learn richer and more diverse features, improving performance in various tasks.

2.3.5. Inception
Szegedy *et al.* [36], Inception is a CNN architecture discovered in 2014 by a research team at Google. Inception is renowned for its utilization of Inception modules. These modules encompass the parallel application of convolutional filters with varying sizes on the input layer, followed by the combination of the obtained results. This approach allows the network to efficiently extract features from multiple spatial scales in an image. Inception is a highly successful architecture that has been widely utilized in various computer vision tasks, including object detection, image classification, and image segmentation.

2.3.6. Xception
Xception is a CNN architecture developed Chollet [37], a data scientist at Google, and introduced in 2017. Xception, short for “Extreme Inception,” is an enhanced version of the Inception module concept initially introduced in the GoogLeNet architecture. Xception carries the idea of separating spatial convolution and channel convolution into two distinct operations. This approach aims to improve network efficiency and performance by reducing the number of required parameters. By separating spatial and channel convolution, Xception allows the network to learn spatial and channel representations independently. Xception has demonstrated high proficiency in a range of computer vision tasks, including image classification and object detection. Furthermore, it has managed to maintain a high level of efficiency throughout.

2.4. Data augmentation
The diversity and quantity of training data are enhanced using data augmentation, a technique that generates modified versions of existing images [38]. This approach reduces overfitting and enhances the model’s ability to generalize [39]. Table 1 shows the configuration of data augmentation used in this study.

Table 1. Data augmentation

Data augmentation	Configuration	Description
Rescale	1/255	Normalize the pixel values of images from the range [0, 255] to the range [0, 1]
Zoom range	0.1	The images can be zoomed in by up to 10% or zoomed out by up to 10%
Rotation range	0.1	The rotation angle can be any value between -0.1 and 0.1 radians
Width shift range	0.1	The horizontal shift ranges between -0.1 and 0.1 times the width of the image
Height shift range	0.1	The vertical shift ranges between -0.1 and 0.1 times the width of the image

2.5. Hyperparameters
In deep learning describe parameters that are not directly learned by the model during the training process but must be determined before the training process begins [40].

- Learning rate: determines how many steps are taken in a given direction to update the model weights during training.
- Batch size: the batch size refers to the number of samples used in each training iteration. It directly impacts the frequency of model weight updates and the speed of the training process.
- Epochs: the number of complete iterations or rounds through the entire training data set used during the training process.
- Optimizer: the algorithm used for adjusting model weights is based on the gradient of the loss function. Some examples of optimizers include Adam, stochastic gradient descent (SGD), and RMSprop.

2.6. Performance evaluation

To assess the classification performance, a confusion matrix has been obtained. The confusion matrix is an important performance evaluation tool in classification that allows us to visualize the model performance in more detail [41]. The confusion matrix provides information on how the predicted results of the classification model align with the actual values of the observed data in Figure 3. Components of the confusion matrix:

– TP: the model correctly predicts the number of positive samples.
– TN: the model accurately predicted the number of negative samples.
– FP: the model incorrectly predicted the number of samples as positive (false alarm).
– FN: the number of samples that the model predicted incorrectly as negative (Miss).

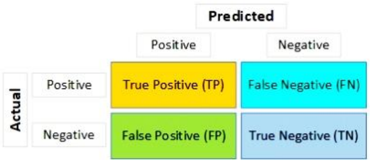


Figure 3. Confusion matrix

From these components, we can calculate several important classification evaluation metrics:

– The accuracy of the model's predictions, which is calculated using (5), is represented as a percentage of correctly predicted samples.

$$Acc = \frac{(TP+TN)}{(TP+TN+FP+FN)} \tag{5}$$

– Precision is calculated as (6) and represents the percentage of positive samples accurately predicted by the model.

$$Pre = \frac{TP}{(TP+FP)} \tag{6}$$

– Recall, also referred to as sensitivity or true positive rate, represents the percentage of positive samples that the model accurately predicts. It can be calculated using (7):

$$Recall = \frac{TP}{(TP+FN)} \tag{7}$$

– Specificity, also known as the true negative rate, refers to the percentage of negative samples that are accurately predicted by the model. It is calculated using (8):

$$Specificity = \frac{TN}{(TN+FP)} \tag{8}$$

– F1 Score, it is calculated as the harmonic mean of precision and recall, and can be represented by (9):

$$F1\ Score = \frac{2*(Precision * Recall)}{(Precision + Recall)} \tag{9}$$

3. RESULTS AND DISCUSSION

3.1. Experiment set-up

The experiments were conducted using the Python programming language, with libraries such as OpenCV, Scikit-Learn, TensorFlow, and Keras. The experiments were run on a PC with the following specifications: Intel Core i7 9th Gen processor, 16 GB DDR4 RAM, and an NVIDIA GeForce GTX 1660 Ti GPU. The input images were in RGB format with a size of 224×224 pixels, and the training process was set for 50 epochs. The ImageDataGenerator was utilized to augment the training and validation data through techniques like rescaling (normalization), zooming, rotating, and shifting both horizontally and vertically. The Adam optimizer was used with a learning rate of 0.00005. The batch size was set to 32. Callbacks like EarlyStopping and ReduceLROnPlateau were employed to prevent overfitting and adjust the learning rate during training. The hyperparameter configurations are summarized in Table 2.

Table 2. Hyperparameter configuration

Hyperparameter	Configuration	Description
Batch size	32	The model processes 32 samples per training iteration
Max Epoch	50	The model is trained for 50 iterations over the dataset
Loss function	Categorical cross-entropy	Used for multi-class classification tasks
Dropout rate	0.1	10% of the neurons are randomly deactivated to prevent overfitting
Optimizer	Adam	An optimization algorithm used for training
Learning rate	0.00005	Controls the step size during the optimization process

Seven transfer learning models were employed in this study: VGG16, VGG19, MobileNet, DenseNet201, ResNet152V2, InceptionV3, and Xception. The goal was to determine which model performs best in classifying the ripeness of palm oil fruits. The input images were processed by the transfer learning models, which used a GlobalAveragePooling2D layer to extract feature vectors. These vectors were then passed through a dense layer with 128 units and ReLU activation, followed by a dropout layer to mitigate overfitting, and finally, a dense output layer with Softmax activation for multiclass classification. The base models were set to untrainable (trainable=False) to retain the learned weights. The architecture of the proposed model is illustrated in Figure 4.

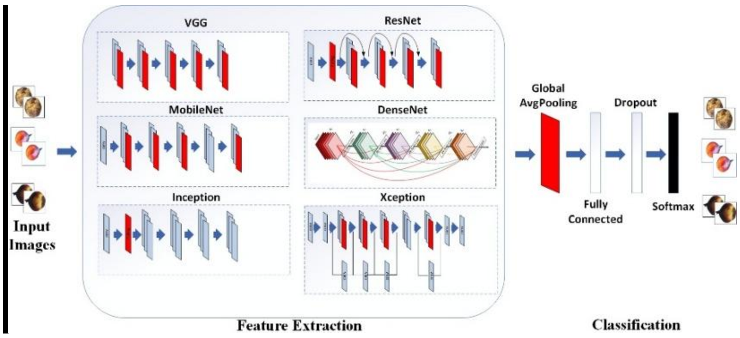


Figure 4. Proposed model experiment

3.2. Result

Figures 5 and 6 show the loss and accuracy graphs for each transfer learning model. The loss graphs indicate how well the models minimized their loss values during training. A decrease in loss over multiple epochs suggests effective learning, while an increase may indicate overfitting. Conversely, the accuracy graphs show the models' ability to make correct predictions. An increase in accuracy across epochs reflects the models' improved prediction capabilities. However, a plateau or decrease in accuracy might suggest that the models have reached their performance limits or are overfitting.

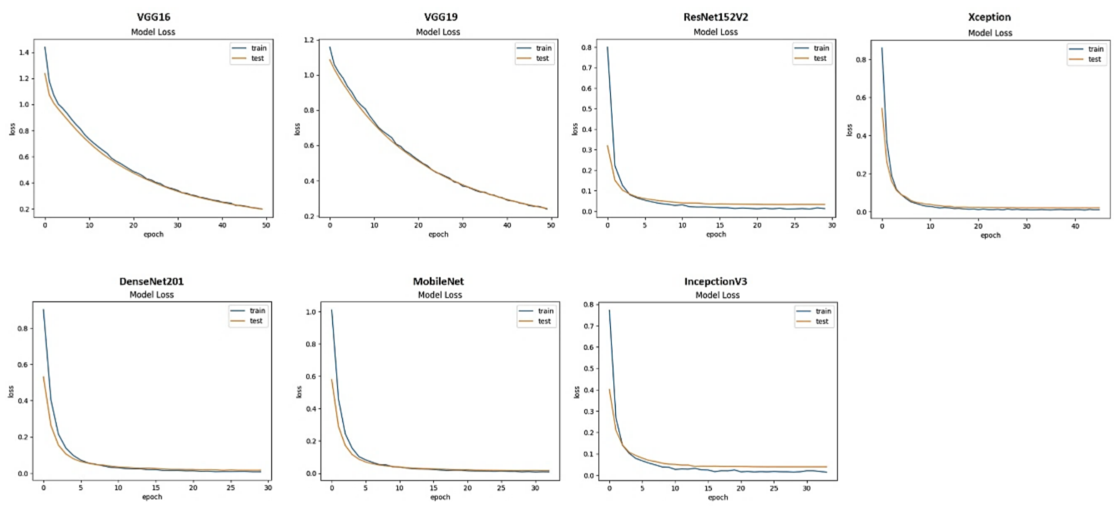


Figure 5. Loss function for each transfer learning model

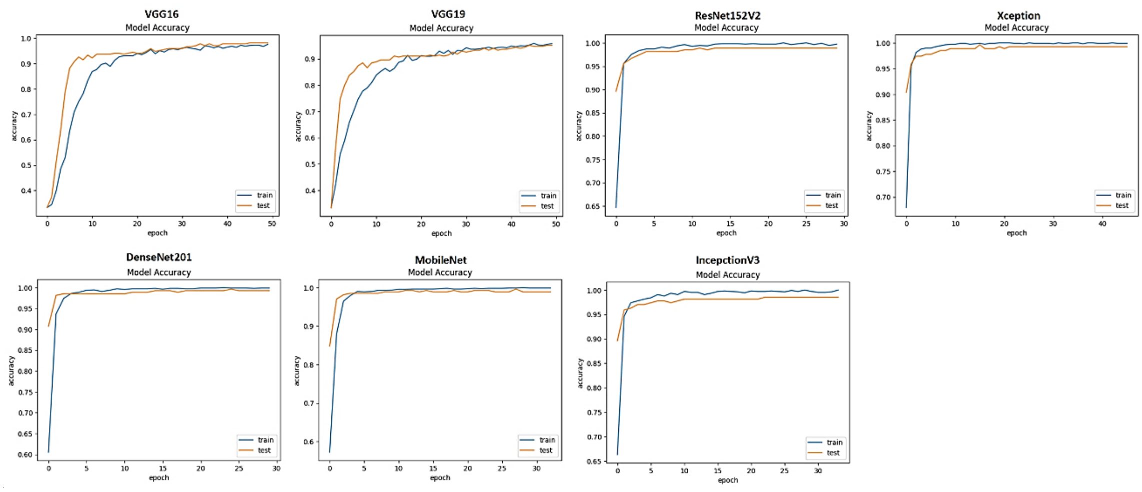


Figure 6. Accuracy for each transfer learning model

The confusion matrices in Figure 7 reveal the classification accuracy for each class across the models. VGG16 and VGG19 performed well in classifying “rotten” and “ripe” classes but struggled slightly with the “raw” class. ResNet152V2, DenseNet201, and MobileNet achieved perfect classification across all classes, while InceptionV3 and Xception demonstrated strong performance with minimal errors, particularly in the “raw” class.

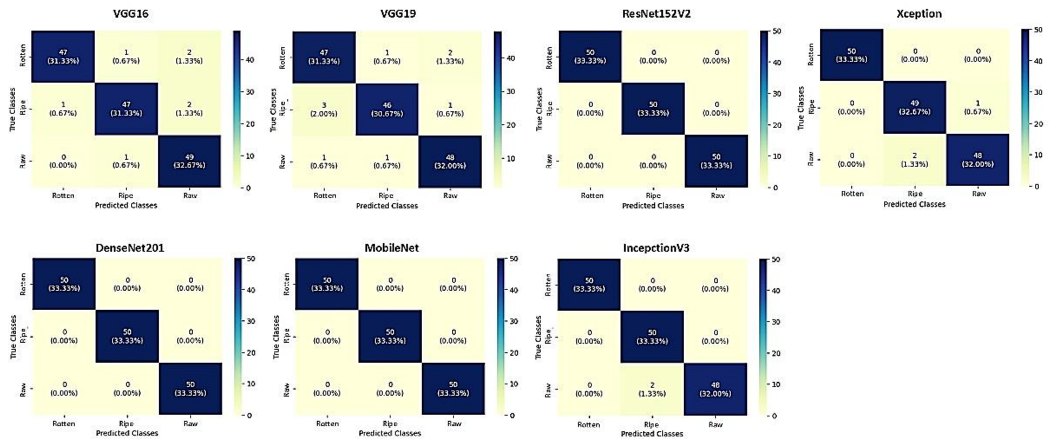


Figure 7. Confusion matrix for each transfer learning model

Table 3 shows a comparison of model performance on validation and test data. The ResNet152V2, DenseNet201, MobileNet, and Xception models perform exceptionally well in both validation and testing phases, with accuracies approaching or reaching 1.00. The VGG16, VGG19, and InceptionV3 models also demonstrate good performance, although not as high as the aforementioned models. Notably, the InceptionV3 model shows a significant discrepancy in loss rates between validation and test data, despite some models achieving perfect accuracy on test data (ResNet152V2, DenseNet201, MobileNet, and Xception). ResNet152V2 exhibits a very low loss value (0.0135) and the highest accuracy (over 99%) on validation data. Other models also perform well, with relatively low loss values and accuracy exceeding 0.95. In terms of recall, precision, and F1 score, most models achieve high values, with some like ResNet152V2, DenseNet201, MobileNet, and InceptionV3 reaching perfect scores. This indicates that these models have a strong capability to classify accurately and consistently. While ResNet152V2 stands out as the best-performing model in terms of loss and accuracy, other models such as DenseNet201, MobileNet, and InceptionV3 also show very good performance in precision, recall, and F1 score. Models that achieve perfect accuracy (1.00) on test data demonstrate excellent generalization, suggesting that they effectively identify patterns beyond the training data. Additionally, model efficiency and complexity are crucial considerations, especially for real-time applications or environments with limited resources, such as mobile devices. Models like MobileNet, with fewer parameters and lower complexity, may offer better efficiency and resource utilization.

Table 3. Comparison of each transfer learning model

Model	Validation		Testing		Precision	Recall	F-1 Score
	Loss	Acc	Loss	Acc			
VGG16	0.1967	0.9814	0.2112	0.9533	0.95	0.95	0.95
VGG19	0.2413	0.9518	0.2720	0.9399	0.94	0.94	0.94
ResNet152V2	0.0316	0.9888	0.0135	1.0000	1.00	1.00	1.00
DenseNet201	0.0167	0.9962	0.0179	1.0000	1.00	1.00	1.00
MobileNet	0.0160	0.9962	0.0221	1.0000	1.00	1.00	1.00
InceptionV3	0.0392	0.9851	0.0474	0.9866	0.99	0.99	0.99
Xception	0.0204	0.9925	0.0487	0.9800	0.98	0.98	0.98

The models ResNet152V2, DenseNet201, MobileNet, and Xception demonstrated superior performance with high accuracy and low loss values in both validation and test data. ResNet152V2 stood out with the highest accuracy and lowest loss, making it the best model in this study. However, to avoid overfitting, it is important to carry out further validation. This evaluation can be done by cross-validation [42].

The cross-validation results in Table 4 show that ResNet152V2, DenseNet201, and MobileNet are consistent, achieving high accuracy and low loss across multiple folds. All models reached perfect accuracy (1.0) on training data, indicating no overfitting. On validation data, ResNet152V2 consistently achieved high performance with low loss and high accuracy. DenseNet201 also performed well, showing similar low loss and high accuracy. MobileNet had good results but showed some variation in loss and accuracy. Generally, models with less variation between folds and lower loss values tend to predict new data better. ResNet152V2 and DenseNet201 demonstrated notable consistency and performance across folds.

Table 4. Evaluation models with cross-validation

Model	Fold	Training		Validation	
		Loss	Accuracy	Loss	Accuracy
ResNet152V2	1	0.000997	1.000.000	0.014198	0.99537
	2	0.001352	1.000.000	0.035341	0.986111
	3	0.00168	1.000.000	0.006712	0.99537
	4	0.000835	1.000.000	0.076519	0.986111
	5	0.003803	0.998843	0.011923	0.99537
DenseNet	1	0.001741	1.000.000	0.005908	0.99537
	2	0.001147	1.000.000	0.014081	0.990741
	3	0.000999	1.000.000	0.011456	0.99537
	4	0.002948	1.000.000	0.025724	0.990741
	5	0.00042	1.000.000	0.000492	1.000.000
MobileNet	1	0.001381	1.000.000	0.001107	1.000.000
	2	0.001988	0.998843	0.00354	1.000.000
	3	0.001423	1.000.000	0.01809	0.990741
	4	0.002221	1.000.000	0.022691	0.990741
	5	0.001852	1.000.000	0.004324	1.000.000

3.3. Discussion

Based on the previous studies that have been discussed, it is clear that while significant progress has been made, there are still opportunities to improve the accuracy of identifying palm fruit ripeness. The use of appropriate transfer learning models, coupled with effective data augmentation techniques and optimal hyperparameters, can potentially lead to better results. In this context, our research significantly outperformed previous studies by leveraging advanced transfer learning models and carefully optimized hyperparameters. The application of CNN-based architectures such as VGGNet, ResNet, MobileNet, DenseNet, Inception, and Xception has enabled us to achieve superior classification accuracy. The combination of these powerful models with our dataset of images, sourced directly from community farmers' plantations in South Sumatra, Indonesia, has resulted in a more accurate and robust identification system for palm oil fruit ripeness. Our findings underscore the importance of model selection and optimization in improving the performance of automated classification systems, setting a new benchmark in the field of palm oil fruit ripeness identification.

This study successfully utilized deep learning-based transfer learning models to classify palm oil fruit ripeness. ResNet152V2 was the most accurate model on test data, with perfect accuracy (1.0000) and the lowest validation loss (0.0135). DenseNet201, MobileNet, and InceptionV3 also showed strong performance with accuracy above 0.99 and validation loss below 0.04. The cross-validation results further highlighted ResNet152V2's consistent performance across folds, solidifying its status as the best-performing model. DenseNet201 and MobileNet also demonstrated consistent and reliable results, making them suitable alternatives. While the results demonstrate high accuracy and strong generalization across different models, there are limitations that may impact the findings. First, the dataset used, although representative, may not capture the full variability of palm oil fruit conditions in different regions and under varying environmental factors. Additionally, while cross-validation was used to assess model stability, further research with larger and more diverse datasets, as well as real-world testing in operational settings, is necessary to confirm the robustness and practicality of these models. Finally, the study did not explore the impact of different data augmentation techniques or hyperparameter tuning in depth, which could potentially enhance the models' performance further.

4. CONCLUSION

This research demonstrates that deep learning models based on transfer learning, particularly ResNet152V2, DenseNet201, and MobileNet, offer substantial benefits in the classification of palm oil fruit ripeness. Among these, ResNet152V2 exhibited the highest accuracy and the lowest validation loss during initial evaluations, as well as consistent and stable performance across cross-validation trials. DenseNet201 and MobileNet also showed robust and reliable performance, making them strong candidates for this classification task. These findings highlight the potential of transfer learning models to revolutionize the process of palm oil fruit ripeness identification, providing a more accurate and reliable alternative to traditional methods. The implementation of such models in practical applications could greatly enhance the efficiency of palm oil harvesting, contributing to better yield management and quality control. Looking ahead, further research could explore the impact of employing alternative data augmentation techniques and conduct more extensive data collection efforts. A broader dataset, encompassing a wider range of growing conditions, maturity stages, and environmental factors, would allow for the development of models with even greater generalization capabilities. By doing so, the models could better handle the inherent variability found in palm oil fruit and achieve more accurate classifications in real-world scenarios. Moreover, future studies could investigate the integration of these models into automated systems for real-time ripeness detection in palm oil plantations. Such systems could significantly reduce the labor required for manual classification and improve decision-making processes in the palm oil industry, ultimately leading to higher productivity and sustainability.

ACKNOWLEDGMENT

We extend our heartfelt thanks to Universitas Bina Insan for providing the resources and support necessary for this study. Special thanks to our colleagues and mentors in the Department of Computer System Engineering for their invaluable guidance and feedback throughout the research process. We also wish to acknowledge the contributions of the dataset providers and the palm oil plantation community in South Sumatra, Indonesia, whose support and cooperation were instrumental in acquiring the data used in this study. This research was conducted with self-funding.

REFERENCES

[1] D. S. Wilcove and L. P. Koh, "Addressing the threats to biodiversity from oil-palm agriculture," *Biodiversity and Conservation*, vol. 19, no. 4, pp. 999–1007, Apr. 2010, doi: 10.1007/s10531-009-9760-x.

[2] A. W. Setiawan, R. Mengko, A. P. H. Putri, D. Danudirdjo, and A. R. Ananda, "Classification of Palm Oil Fresh Fruit Bunch using Multiband Optical Sensors," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 9, no. 4, pp. 2386~2393, Aug. 2019, doi: 10.11591/ijece.v9i4.pp2386-2393.

[3] N. Fadilah, J. Mohamad-Saleh, Z. A. Halim, H. Ibrahim, and S. S. S. Ali, "Intelligent Color Vision System for Ripeness Classification of Oil Palm Fresh Fruit Bunch," *Sensors*, vol. 12, no. 10, pp. 14179–14195, Oct. 2012, doi: 10.3390/s121014179.

[4] O. M. Bensaeed, A. M. Shariff, A. B. Mahmud, H. Shafri, and M. Alfatni, "Oil palm fruit grading using a hyperspectral device and machine learning algorithm," *IOP Conference Series: Earth and Environmental Science*, vol. 20, p. 012017, Jun. 2014, doi: 10.1088/1755-1315/20/1/012017.

[5] A. Y. Saleh and E. Liansitim, "Palm oil classification using deep learning," *Science in Information Technology Letters*, vol. 1, no. 1, pp. 1–8, May 2020, doi: 10.31763/sitech.v1i1.1.

[6] Suharjito *et al.*, "Annotated Datasets of Oil Palm Fruit Bunch Piles for Ripeness Grading Using Deep Learning," *Scientific Data*, vol. 10, no. 1, p. 72, Feb. 2023, doi: 10.1038/s41597-023-01958-x.

[7] Z. Y. Wong, W. J. Chew, and S. K. Phang, "Computer vision algorithm development for classification of palm fruit ripeness," p. 030012, May 2020, doi: 10.1063/5.0002188.

[8] A. Septiarini, H. R. Hatta, H. Hamdani, A. Oktavia, A. A. Kasim, and S. Suyanto, "Maturity Grading of Oil Palm Fresh Fruit Bunches Based on a Machine Learning Approach," in *2020 Fifth International Conference on Informatics and Computing (ICIC)*, pp. 1–4, Nov. 2020, doi: 10.1109/ICIC50835.2020.9288603.

[9] T. Raj, F. H. Hashim, A. B. Huddin, A. Hussain, M. F. Ibrahim, and P. M. Abdul, "Classification of oil palm fresh fruit maturity based on carotene content from Raman spectra," *Scientific Reports*, vol. 11, no. 1, p. 18315, Sep. 2021, doi: 10.1038/s41598-021-97857-5.

[10] S. Deepak and P. M. Ameer, "Brain tumor classification using deep CNN features via transfer learning," *Computers in Biology and Medicine*, vol. 111, p. 103345, Aug. 2019, doi: 10.1016/j.compbiomed.2019.103345.

[11] E. Baykal, H. Dogan, M. E. Ercin, S. Ersoz, and M. Ekinci, "Transfer learning with pre-trained deep convolutional neural networks for serous cell classification," *Multimedia Tools and Applications*, vol. 79, no. 21–22, pp. 15593–15611, Jun. 2020, doi: 10.1007/s11042-019-07821-9.

[12] Z. N. K. Swati *et al.*, "Brain tumor classification for MR images using transfer learning and fine-tuning," *Computerized Medical Imaging and Graphics*, vol. 75, pp. 34–46, July 2019, doi: 10.1016/j.compmedimag.2019.05.001.

[13] A. Rehman, S. Naz, M. I. Razzak, F. Akram, and M. Imran, "A Deep Learning-Based Framework for Automatic Brain Tumors Classification Using Transfer Learning," *Circuits, Systems, and Signal Processing*, vol. 39, no. 2, pp. 757–775, Feb. 2020, doi: 10.1007/s00034-019-01246-3.

[14] R. Singh, T. Ahmed, A. Kumar, A. K. Singh, A. K. Pandey, and S. K. Singh, "Imbalanced Breast Cancer Classification Using Transfer Learning," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 18, no. 1, pp. 83–93, Jan. 2021, doi: 10.1109/TCBB.2020.2980831.

[15] M. A. Wakili *et al.*, "Classification of Breast Cancer Histopathological Images Using DenseNet and Transfer Learning," *Computational Intelligence and Neuroscience*, vol. 2022, pp. 1–31, Oct. 2022, doi: 10.1155/2022/8904768.

[16] T. S. Pan, H. C. Huang, J. C. Lee, and C. H. Chen, "Multi-scale ResNet for real-time underwater object detection," *Signal, Image and Video Processing*, vol. 15, no. 5, pp. 941–949, Jul. 2021, doi: 10.1007/s11760-020-01818-w.

[17] H. Pan, Z. Pang, Y. Wang, Y. Wang, and L. Chen, "A New Image Recognition and Classification Method Combining Transfer Learning Algorithm and MobileNet Model for Welding Defects," *IEEE Access*, vol. 8, pp. 119951–119960, Jun. 2020, doi: 10.1109/ACCESS.2020.3005450.

[18] C. Wang *et al.*, "Pulmonary Image Classification Based on Inception-v3 Transfer Learning Model," *IEEE Access*, vol. 7, pp. 146533–146541, Oct. 2019, doi: 10.1109/ACCESS.2019.2946000.

[19] W. W. Lo, X. Yang, and Y. Wang, "An Xception Convolutional Neural Network for Malware Classification with Transfer Learning," in *2019 10th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*, pp. 1–5, Jun. 2019, doi: 10.1109/NTMS.2019.8763852.

[20] A. Septiarini, H. Hamdani, H. R. Hatta, and K. Anwar, "Automatic image segmentation of oil palm fruits by applying the contour-based approach," *Scientia Horticulturae*, vol. 261, p. 108939, Feb. 2020, doi: 10.1016/j.scienta.2019.108939.

[21] A. Septiarini, A. Sunyoto, H. Hamdani, A. A. Kasim, F. Utaminingrum, and H. R. Hatta, "Machine vision for the maturity classification of oil palm fresh fruit bunches based on color and texture features," *Scientia Horticulturae*, vol. 286, p. 110245, Aug. 2021, doi: 10.1016/j.scienta.2021.110245.

[22] M. S. M. Alfatni, A. R. M. Shariff, O. M. B. Saaed, A. M. Albhbah, and A. Mustapha, "Colour Feature Extraction Techniques for Real Time System of Oil Palm Fresh Fruit Bunch Maturity Grading," *IOP Conference Series: Earth and Environmental Science*, vol. 540, no. 1, p. 012092, Jul. 2020, doi: 10.1088/1755-1315/540/1/012092.

[23] M. S. M. Alfatni, S. Khairunniza-Bejo, M. H. B. Marhaban, O. M. B. Saaed, A. Mustapha, and A. R. M. Shariff, "Towards a Real-Time Oil Palm Fruit Maturity System Using Supervised Classifiers Based on Feature Analysis," *Agriculture*, vol. 12, no. 9, p. 1461, Sep. 2022, doi: 10.3390/agriculture12091461.

[24] M. Y. M. A. Mansour, K. D. Dambul, and K. Y. Choo, "Object Detection Algorithms for Ripeness Classification of Oil Palm Fresh Fruit Bunch," *International Journal of Technology*, vol. 13, no. 6, pp. 1326-1335, Nov. 2022, doi: 10.14716/ijtech.v13i6.5932.

[25] M. Shiddiq, F. Candra, B. Anand, and M. F. Rabin, "Neural network with k-fold cross validation for oil palm fruit ripeness prediction," *Telecommunication Computing Electronics and Control (TELKOMNIKA)*, vol. 22, no. 1, pp. 164~174, Feb. 2024, doi: 10.12928/telkomnika.v22i1.24845.

[26] A. Dhillon and G. K. Verma, "Convolutional neural network: a review of models, methodologies and applications to object detection," *Progress in Artificial Intelligence*, vol. 9, no. 2, pp. 85–112, Jun. 2020, doi: 10.1007/s13748-019-00203-0.

[27] Y. Sun, B. Xue, M. Zhang, and G. G. Yen, "Evolving Deep Convolutional Neural Networks for Image Classification," *IEEE Transactions on Evolutionary Computation*, vol. 24, no. 2, pp. 394–407, Apr. 2020, doi: 10.1109/TEVC.2019.2916183.

[28] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in *2017 International Conference on Engineering and Technology (ICET)*, pp. 1–6, Aug. 2017, doi: 10.1109/ICEngTechnol.2017.8308186.

[29] M. Gao *et al.*, "Holistic classification of CT attenuation patterns for interstitial lung diseases via deep convolutional neural networks," *Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization*, vol. 6, no. 1, pp. 1–6, Jan. 2018, doi: 10.1080/21681163.2015.1124249.

[30] S. S. Udmale, S. K. Singh, R. Singh, and A. K. Sangaiah, "Multi-Fault Bearing Classification Using Sensors and ConvNet-Based Transfer Learning Approach," *IEEE Sensors Journal*, vol. 20, no. 3, pp. 1433–1444, Feb. 2020, doi: 10.1109/JSEN.2019.2947026.

[31] 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*, pp. 1-14, 2015, doi: 10.48550/arXiv.1409.1556.

[32] S. Targ, D. Almeida, and K. Lyman, "Resnet in Resnet: Generalizing Residual Architectures," *in ICLR 2016*, pp. 1-7, 2016, doi: 10.48550/arXiv.1603.08029.

[33] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, Jun. 2016, doi: 10.1109/CVPR.2016.90.

[34] A. G. Howard *et al.*, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," *arXiv*, 2017, doi: 10.48550/arXiv.1704.04861.

[35] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely Connected Convolutional Networks," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 39, no. 9, pp. 1442–1446, 1978, doi: 10.48550/arXiv.1608.06993.

[36] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, and S. Reed, "Going Deeper with Convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–9, 2015, doi: 10.1109/CVPR.2015.7298594.

[37] F. Chollet, "Xception: Deep Learning with Depthwise Separable Convolutions," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1800–1807, Jul. 2017, doi: 10.1109/CVPR.2017.195.

[38] C. Shorten and T. M. Khoshgoftaar, "A survey on Image Data Augmentation for Deep Learning," *Journal of Big Data*, vol. 6, no. 60, Jul. 2019, doi: 10.1186/s40537-019-0197-0.

[39] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, May 2017, doi: 10.1145/3065386.

[40] T. Yu and H. Zhu, "Hyper-Parameter Optimization: A Review of Algorithms and Applications," *arXiv*, Mar. 2020, doi: 10.48550/arXiv.2003.05689.

[41] L. S. Bernardo, R. Damaševičius, V. H. C. de Albuquerque, and R. Maskeliūnas, "A hybrid two-stage SqueezeNet and support vector machine system for Parkinson's disease detection based on handwritten spiral patterns," *International Journal of Applied Mathematics and Computer Science*, vol. 31, no. 4, pp. 549–561, 2021, doi: 10.34768/amcs-2021-0037.

[42] V. Vakharia and R. Gujar, "Prediction of compressive strength and portland cement composition using cross-validation and feature ranking techniques," *Construction and Building Materials*, vol. 225, pp. 292–301, Nov. 2019, doi: 10.1016/j.conbuildmat.2019.07.224.

BIOGRAPHIES OF AUTHORS

Rudi Kurniawan is currently a lecturer at Department of Computer System Engineering, Faculty of Engineering Science, Universitas Bina Insan, Lubuklinggau, Indonesia. He received Bachelor of Engineering in Electrical Engineering from Gadjah Mada University, Indonesia, Magister of Computer in Informatics Engineering from Bina Darma University, Indonesia, and Doctoral in Engineering Science from Universitas Sriwijaya, Indonesia. His research interest is image processing, machine learning, and pattern recognition. He can be contacted at email: rudi.kurniawan@univbinainsan.ac.id.

Samsuryadi is an Associate Professor at the Department of Informatics Engineering, Universitas Sriwijaya, Indonesia. He recieved Bachelor of Science in Mathematics from Universitas Sriwijaya, Indonesia, a Magister of Computer Science from Gadjah Mada University, Indonesia, and Ph.D. in Computer Science, Universiti Teknologi Malaysia, Malaysia. His research interest is image processing, machine learning, and pattern recognition. He can be contacted at email: samsuryadi@unsri.ac.id.

Fatma Susilawati Mohamad is an Associate Professor at the Department of Information Technology, University Sultan Zainal Abidin, Kuala Terengganu, Malaysia. She received a Bachelor of Science in Information System Management from Oklahoma, USA, MIT Computer Science from Universiti Kebangsaan Malaysia, and Ph.D. in Computer Science, from Universiti Teknologi Malaysia. Her research interest is feature extraction methods, pattern mining, pattern analysis, and object classification. She can be contacted at email: fatma@unisza.edu.my.

Harma Oktafia Lingga Wijaya is currently lecturer at Department of Information System, Faculty of Engineering Science, Universitas Bina Insan, Indonesia. She received Bachelor of Computer from Universitas Bina Insan, Indonesia and Magister of Computer in Informatics Engineering from Bina Darma University, Indonesia. Her research interest is data science, machine learning, and pattern recognition. She can be contacted at email: harmaoktafialingga@univbinainsan.ac.id.

Budi Santoso is currently lecturer at Department of Informatics, Faculty of Engineering Science, Universitas Bina Insan, Indonesia. He received Bachelor of Computer from Universitas Bina Darma, Indonesia; Magister of Computer in Informatics Engineering from Bina Darma University, Indonesia. His research interest is data science, machine learning, and pattern recognition. He can be contacted at email: budisantoso@univbinainsan.ac.id.