SINERGI Vol. 29, No. 1, February 2025: 207-220
http://publikasi.mercubuana.ac.id/index.php/sinergi
http://doi.org/10.22441/sinergi.2025.1.019

Classification of palm oil fruit ripeness based on AlexNet deep Convolutional Neural Network

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

Abstract
The palm oil industry faces significant challenges in accurately classifying fruit ripeness, which is crucial for optimizing yield, quality, and profitability. Manual methods are slow and prone to errors, leading to inefficiencies and increased costs. Deep Learning, particularly the AlexNet architecture, has succeeded in image classification tasks and offers a promising solution. This study explores the implementation of AlexNet to improve the efficiency and accuracy of palm oil fruit maturity classification, thereby reducing costs and production time. We employed a dataset of 1500 images of palm oil fruits, meticulously categorized into three classes: raw, ripe, and rotten. The experimental setup involved training AlexNet and comparing its performance with a conventional Convolutional Neural Network (CNN). The results demonstrated that AlexNet significantly outperforms the traditional CNN, achieving a validation loss of 0.0261 and an accuracy of 0.9962, compared to the CNN's validation loss of 0.0377 and accuracy of 0.9925. Furthermore, AlexNet achieved superior precision, recall, and F-1 scores, each reaching 0.99, while the CNN scores were 0.98. These findings suggest that adopting AlexNet can enhance the palm oil industry's operational efficiency and product quality. The improved classification accuracy ensures that fruits are harvested at optimal ripeness, leading to better oil yield and quality. Reducing classification errors and manual labor can also lead to substantial cost savings and increased profitability. This study underscores the potential of advanced deep learning models like AlexNet in revolutionizing agricultural practices and improving industrial outcomes.

Keywords:
AlexNet;
Classification;
Deep learning;
Fruit ripeness;
Image classification;
Palm oil industry;

Article History:
Received: May 1, 2024
Revised: July 4, 2024
Accepted: July 24, 2024
Published: February 4, 2025

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

INTRODUCTION

One of the rapidly developing agro-industries in Indonesia is the palm oil industry [1]. The palm oil industry plays a significant role in the global economy, especially in 3producing palm oil, which is widely used in the food, cosmetics, and fuel industries. Since the ripeness of the palm fruit impacts the quality and quantity of the palm oil produced, classifying the ripeness of the palm fruit is an important step in the harvesting and processing process [2].

Traditionally, palm oil fruit ripeness has been classified manually by human workers. However, this method is often time-consuming, costly, and susceptible to inaccuracies due to individual subjectivity. As a result, current manual techniques for ripeness classification could be more efficient and prone to human error, rendering them unreliable for large-scale applications.

Numerous studies have explored the classification of palm oil fruits using machine learning and deep learning techniques, such as artificial neural networks (ANN) [3], CNN [4][5], [6], and linear discriminant analysis (LDA) [7]. However, many of these studies still need to fully exploit the potential of deep learning models, which have shown great promise in solving image classification problems.

Deep learning, particularly Convolutional Neural Networks (CNN), has proven to be highly effective in image classification tasks [8][9]. By leveraging deep CNNs, complex patterns related to oil palm fruit maturity can be learned from the collected image data, which is essential for accurate classification. Several studies have successfully applied deep learning to palm oil fruit images [10, 11, 12].

AlexNet is a highly influential CNN-based image-processing architecture that has significantly advanced the field of image classification [13]. Developed by Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton, AlexNet won the ImageNet competition in 2012, demonstrating remarkable performance in image classification tasks. This architecture has laid the foundation for many sophisticated CNN architectures developed later and represents a significant milestone in deep learning for imaging [14][15].

Applying Deep CNN AlexNet to palm oil fruit classification offers several significant benefits:

1. High classification accuracy: AlexNet has demonstrated exceptional accuracy in classifying images. By applying AlexNet to palm oil fruit images, we can expect significant improvements in the accuracy of fruit ripeness classification, which is critical for harvesting and processing.
2. More accurate ripeness detection: Classifying palm oil fruit's ripeness is challenging due to complex variations in fruit color, texture, and shape. AlexNet's ability to extract deep features from images can lead to more accurate ripeness detection under varying conditions.
3. Increased efficiency in monitoring: Implementing AlexNet's Deep CNN can automate the ripeness classification process, reducing reliance on manual observations that are time-consuming and prone to human error. This will increase the efficiency of fruit ripeness monitoring on a large scale, especially in extensive palm oil fields.
4. Cost and time reduction: Automating palm oil fruit ripeness classification with Deep CNN AlexNet can lower the cost and time required for the harvesting and processing process and minimize losses due to decision errors.

In conclusion, the application of Deep CNN AlexNet in palm oil fruit ripeness classification is poised to significantly enhance productivity, efficiency, and quality in the palm oil sector.

RELATED WORK

Despite significant advancements in the classification of palm oil fruit ripeness using various image processing and machine learning techniques, there remain several gaps that the implementation of advanced deep learning models such as AlexNet can address:

The research by [16] focused on segmenting oil palm fruits using a contour-based approach combined with the Canny algorithm. While this method achieved a reasonable accuracy of 90.13%, it primarily relied on shape and color features and faced noise removal and segmentation accuracy challenges. The contour-based method might need to be more robust to handle the complex variations in the appearance of palm oil fruits under different conditions.

Other research by [17] utilized feature extraction based on color and texture, followed by feature selection using PCA and classification with an artificial neural network (ANN). Although this approach achieved a high accuracy of 98.3%, it heavily depended on handcrafted features, which may not capture the intricate details and variations in the images as effectively as deep learning models can.

In the research conducted by [18], image-processing technology combined with artificial neural networks was used to classify oil palm fruit bunches based on color features. Despite achieving an accuracy rate of approximately 94%, the method's reliance on color features alone might limit its performance under varying lighting conditions and image quality.

Another research study conducted by [19] developed a real-time system using various image-processing techniques to classify the ripeness of Fresh Fruit Bundles (FFB). While the BGLAM+ANN algorithm performed well, with an accuracy rate of over 93%, the system's dependence on multiple feature extraction techniques can be computationally intensive and less scalable compared to a unified deep learning approach.

Research conducted by [20], proposed several object detection algorithms to classify the maturity of fresh oil palm fruit bunches, with YOLOv5 showing promising results with an average precision of 0.842. Although these models demonstrated good performance, they primarily focused on object detection rather than a comprehensive feature extraction and classification approach offered by deep learning architectures like AlexNet. Table 1 summarizes the literature on the classification of oil palm fruit.

AlexNet, a deep Convolutional Neural Network (CNN) architecture, offers several advantages over the methods mentioned above:

1. Automated Feature Extraction: unlike traditional methods that rely on handcrafted features, AlexNet automatically learns and extracts complex features from the raw image data, capturing intricate patterns related to palm oil fruit ripeness.
2. Robustness to Variations: AlexNet's deep learning capabilities enable it to handle variations in fruit color, texture, and shape more effectively, providing more accurate ripeness classification even under different environmental conditions.
3. Scalability and Efficiency: AlexNet's unified architecture streamlines the classification process, making it more scalable and computationally efficient than methods that require multiple feature extraction techniques.
4. Enhanced accuracy by leveraging a large dataset and deep learning, AlexNet can achieve higher accuracy rates, reducing the potential for human error and improving the reliability of the ripeness classification system.

Therefore, the application of AlexNet in palm oil fruit ripeness classification addresses the limitations of previous methods, providing a more robust, scalable, and accurate solution that enhances productivity, efficiency, and quality in the palm oil industry.

METHOD

This research aims to achieve a high-accuracy classification of palm oil fruit images into three categories: raw, ripe, and rotten. Accurate classification is crucial because it directly impacts the quality and efficiency of the harvesting and processing processes in the palm oil industry. Misclassification can lead to premature or delayed harvesting, affecting yield and product quality. The flowchart in Figure 1 shows the research plan for this study.

We employed AlexNet a deep Convolutional Neural Network (CNN) to address these challenges due to its proven effectiveness in image classification tasks. AlexNet's architecture is designed to automatically extract and learn complex features from input images, which is expected to enhance the classification accuracy of palm oil fruit ripeness.

To further improve the classification accuracy, the architecture of the CNN model was carefully designed and modified as necessary. We experimented with different configurations and architectures to determine the setup that provided the best performance for our specific task. This included:

1. Layer Configuration: Adjusting the number of convolutional and fully connected layers.
2. Filter Sizes: Experimenting with different sizes and numbers of filters in convolutional layers.
3. Activation Functions: Using ReLU and other activation functions to optimize the feature extraction process.

We implemented these strategies to achieve a highly accurate and efficient system for classifying palm oil fruit ripeness using AlexNet Deep CNN. The upcoming sub-section will provide a further explanation of each stage.

Material

The data for this study was sourced directly from community farmers' plantations in Durian Remuk Village, District of Muara Beliti, Regency of Musi Rawas, South Sumatra, Indonesia. Durian Remuk Village is predominantly inhabited by palm oil fruit farmers, with an area of approximately 394 hectares.

Table 1. Summary of Literature Review

Author	Preprocessing	Method	Result (%)
[16]	Use of a contour-based approach combined with the Canny algorithm	Segmentation of oil palm fruit using shape and color features	90.13
[17]	Color- and texture-based feature extraction followed by feature selection using PCA	ANN	98.3
[18]	Image processing technology based on color characteristics	ANN	94
[19]	Use of various image processing techniques for real-time FFB maturity classification	BGLAM+ANN algorithm	93
[20]	Proposed several object detection algorithms	YOLOv5	0.84

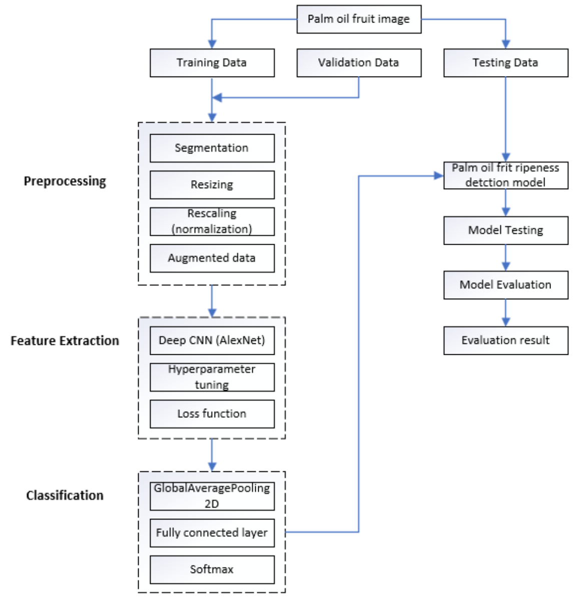

Figure 1. Research Plan

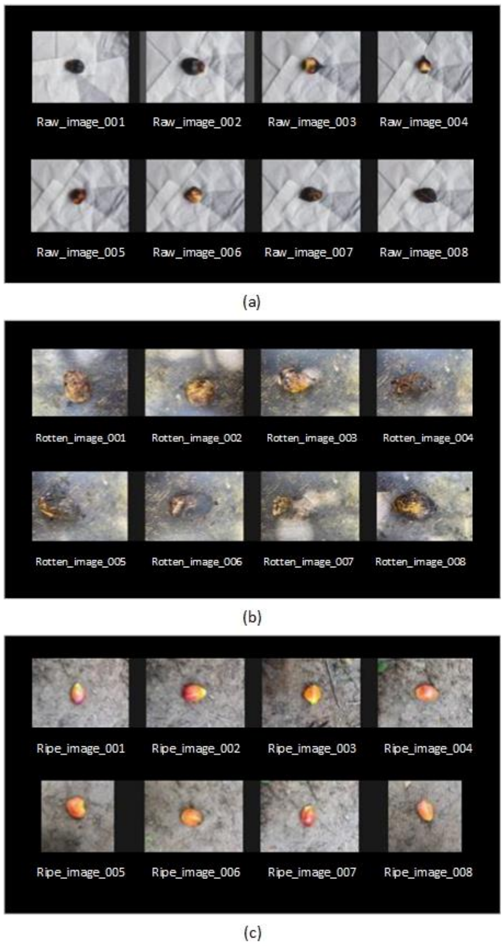

Figure 2. Palm oil ripeness dataset, (a) Raw Palm Oil Fruit, (b) Rotten Palm Oil Fruit, (c) Ripe Palm Oil Fruit

This location was chosen to ensure the relevance and diversity of the dataset, as it represents typical conditions and practices in palm oil farming. Data were collected using a mobile phone camera with a resolution of 128 MP in JPG format to ensure high-quality and detailed images. The high-resolution camera was crucial for capturing detailed features of the palm oil fruits, which are essential for accurate classification. One thousand five hundred images were collected and categorized into three classes: 500 images of ripe palm oil fruits, 500 images of raw palm oil fruits, and 500 images of rotten palm oil fruits. All images were resized with 640x640 pixels. Figure 2 presents sample images from each class.

Methods

Preprocessing

The collected raw data were organized into two main folders: train and test. Each class (ripe, raw, and rotten) contains 450 training and 50 testing images. For validation, 20% of the training data was reserved, resulting in 1080 training images, 270 validation images, and 150 testing images in total. To facilitate the training process, several preprocessing steps were applied:

1. Resizing: All images were resized to 224x224 pixels to ensure uniform input dimensions for the AlexNet model. This standardization helps the model process images more efficiently.
2. Segmentation: The background of each image was segmented using the Hue, Saturation, Value (HSV) color space technique, which isolates the foreground (palm fruit) from the background [21]. This step is crucial for reducing noise and focusing the model on the relevant features of the fruit. Figure 3 illustrates the results of the segmentation process.
3. Normalization (rescale): The pixel values of all images were normalized to a range of 0 to 1. Normalization helps speed up the training process and improves the model's convergence by ensuring that the data distribution is consistent.
4. Augmentation: Data augmentation can be interpreted as an approach used in deep learning to increase the variety and amount of training data from existing images. [28]. Augmentation aims to reduce overfitting, improve model generalization, and make the model more resilient to input data variation [12]. Zoom, rotating, horizontal shifting, and vertical shifting are some of the augmentation methods used in this research.

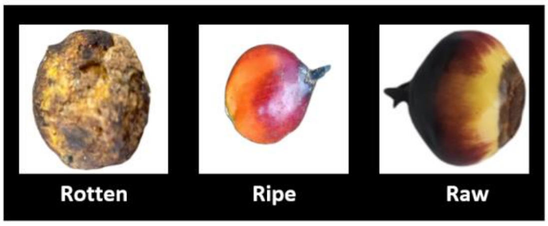

Figure 3. Image Segmentation with HSV

Convolutional Neural Network

Convolutional Neural Networks (CNNs) are a type of artificial neural network architecture specifically designed to process and understand image data through convolutional layers [22], [23]. CNNs are inspired by how the human brain processes visual information [24]. Figure 4 shows the steps of CNN architecture.

Steps in the CNN model:

1. CNN receives images as input, typically of a fixed size.
2. Convolutional layers: These layers apply a series of filters (kernels) to the input image to extract essential features. Each filter detects specific patterns such as edges, textures, or shapes. A filter might detect horizontal edges in the first layer, while more complex patterns like corners or object parts might be detected in deeper layers. ReLU (Rectified Linear Unit) is applied after each convolution to introduce nonlinearity and help the network learn complex patterns.
3. Pooling Layers: Pooling layers reduce the feature maps' spatial dimensions (width and height) while retaining the most essential information. Max pooling selects the maximum value from a patch of the feature map, effectively summarizing the presence of features. Pooling helps reduce overfitting and computational load by decreasing the parameters.
4. Fully connected layers: After a series of convolutional and pooling layers, the neural network's high-level reasoning is done via fully connected layers. The features extracted are converted into a one-dimensional vector and passed through these layers for classification. Activation Functions (ReLU) are commonly used in hidden layers, while softmax is used in the output layer for multi-class classification.
5. Output: The final layer provides the classification results, which are the probabilities of the image belonging to different classes. For example, in a palm oil fruit classification task, the output might be probabilities for the classes raw, ripe, and rotten.

A variety of image processing applications have successfully used CNNs, such as object recognition [25], face detection [26], medical image classification [27], agriculture [28], and other image classification.

ADAM Optimization

ADAM (Adaptive Moment Estimation) is a viral optimization algorithm in deep learning [29]. It combines concepts from stochastic gradient descent (SGD) and momentum algorithms into an adaptive optimizer. Some key ADAM optimizer concepts include Stochastic Gradient Descent (SGD), momentum, first and second moment estimates, learning rate Adjustment, and overfitting prevention.

The ADAM optimizer has proven to be very effective in training deep learning models in diverse applications like natural language processing, image recognition, and sequence modeling [30]. Its key advantage is its adaptive capabilities, which enable fast and stable convergence on a wide range of data types and model architectures.

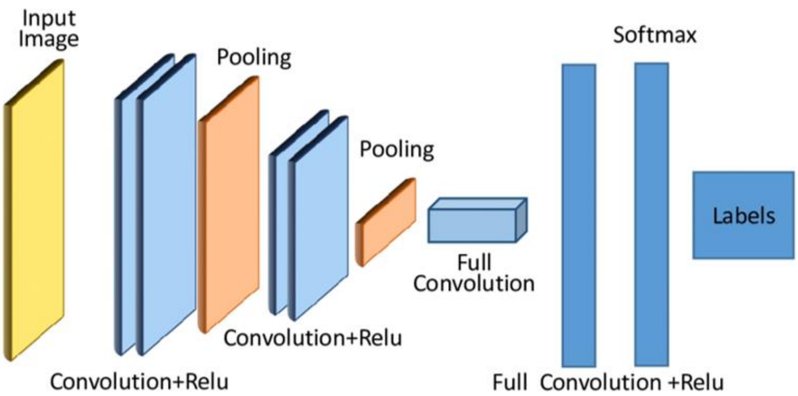

Figure 4. CNN Architecture

ADAM generally achieves higher accuracy and stability across various tasks on benchmark datasets than SGD and RMSProp. [31]. This makes them highly suitable for a wide range of deep-learning tasks.

AlexNet

AlexNet is a CNN architecture best known for winning the ImageNet competition 2012 [32]. This was a significant turning point in the development of deep learning and was one of the models that sparked a surge of interest in the field [15].

The success of AlexNet demonstrates the potential of convolutional neural networks (CNNs) in effectively addressing image classification problems. This architecture has since become the foundation for many more advanced CNN architectures that have been developed.

Table 2 describes the summary of AlexNet architecture with the implementation of TensorFlow. Here is an explanation of the architecture: (1) Input layer: Accepts RGB images with a resolution of 227x227. Convolutional layer: Extracts features from input images using filters of various sizes and numbers; (2) Max Pooling Layers: Reduces the spatial dimensions of images, preserves essential features, and reduces overfitting; (3) Flatten Layer: Converts the 3D output of the last convolution layer to a 1D vector; (4) Fully connected slices: Apply classification based on extracted features, with dropout to prevent overfitting; (5) Output Layer: Generates class probabilities using the Softmax activation function.

Evaluation Metric

A confusion matrix was extracted to evaluate the classification performance (Figure 5). The data in the confusion matrix is an expression of actual and predicted labels by the classifier. Metrics provide an overview of how well the model predicts or classifies data. [33].

Evaluation metrics are beneficial in understanding how well our model predicts new data. Some commonly used evaluation metrics in machine learning are accuracy, F1 score, precision, and recall.

The parameters utilized for comparison consist of accuracy, precision, recall, F1 score, true-positive (TP), true-negative (TN), false-positive (FP), and false-negative (FN) [34][35]. The evaluation metrics are computed using the provided equations outlined in Table 3.

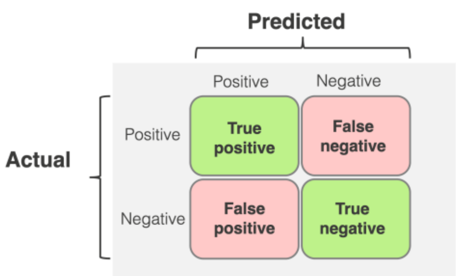

Figure 5. Confusion Matrix

Table 2. Summary of AlexNet Architecture

Layer	Output	Description
Input	(224, 224, 3)	Input layer, RGB image
Conv2D	(54, 54, 96)	96 filters, 11x11 kernel, stride 4
MaxPooling2D	(27, 27, 96)	3x3 pool size, stride 2
Conv2D_1	(17, 17, 256)	256 filters, 5x5 kernel, stride 1
MaxPooling2D_1	(8, 8, 256)	3x3 pool size, stride 2
Conv2D_2	(6, 6, 384)	384 filters, 3x3 kernel, stride 1
Conv2D_3	(4, 4, 384)	384 filters, 3x3 kernel, stride 1
Conv2D_4	(2, 2, 256)	256 filters, 3x3 kernel, stride 1
MaxPooling2D_2	(1, 1, 256)	3x3 pool size, stride 2
Flatten	(256)	Flattening the 3D tensor to 1D
Dense	(4096)	Fully connected layer with 4096 units
Dropout	(4096)	Dropout layer for regularization
Dense_1	(4096)	Fully connected layer with 4096 units
Dropout_1	(4096)	Dropout layer for regularization
Dense_2	(1000)	Fully connected layer with 1000 units
Dropout_2	(1000)	Dropout layer for regularization
Dense_3	(3)	Fully connected layer with 3 output classes (softmax layer)

Table 3. Evaluation Metrics

Evaluation Metrics	Equation
Accuracy	$\frac{TP+TN}{TP+TN+FP+FN}$
Precision	$\frac{TP}{TP+FP}$
Recall	$\frac{TP}{TP+FN}$
F-1 Score	$\frac{Precision * Recall}{Precision + Recall}$

RESULTS AND DISCUSSION

Experiment Set Up

The experiments in this research used the Python programming language and libraries such as OpenCV, Sci-kit Learn, TensorFlow, and Keras. A PC with the following specifications was used for this experiment: CPU processor core i7 gen 9th, DDR4 16 GB, and GPU NVIDIA GeForce GTX 1660 Ti. Testing in this experiment went through 2 scenarios, namely, testing with CNN architecture and AlexNet architecture. The purpose of this test is to see the performance of AlexNet architecture.

All input images were RGB with size 224x224 for both experiments. The network training duration was set as 50 epochs. Image Data Generator is provided to produce enhanced data to train and validate, including rescaling (normalization), zooming, rotating, horizontal shifting, and vertical shifting. Adaptive ADAM optimization was used, with a learning rate of 0.00003. The batch size is 32, and Callbacks, such as EarlyStopping, are used to stop training when there is no further performance improvement. ReduceLROnPlateau is used to reduce the learning rate when model performance has stagnated.

Experiment Set Up with CNN Architecture

Figure 6 shows the architecture of CNN. This architecture consists of several convolutional and other layers. The input layer accepts an image with the dimensions (224, 224, 3) representing image height, width, and color channels (RGB). There are three blocks of convolution, and each block is made up of two layers of convolution. Each convolution layer has a different filter kernel, followed by ReLU activation. These convolution layers are responsible for extracting features from the image. After each convolution block, for each block consisting of two convolution layers, there is a max-pooling layer, which is responsible for reducing the picture dimensions so that the number of parameters processed in the next layer is smaller. After convolution, the output of the final layer will be flattened into a one-dimensional vector with a total of 36864 elements. After the flattening layer, there are two dense layers. The dense layer is responsible for fully connecting each node of the previous layer. The final layer uses a softmax activation function to generate output class probabilities.

A dropout layer is applied after the first dense layer to prevent overfitting by randomly dropping some nodes during training. This model has 2,438,563 parameters, including trainable weight parameters. Because they are all set as trainable, all of these parameters can be trained during training.

An explanation of each layer selection in the CNN architecture is described below:

Block convolution with two convolution layers

1. Each convolution block has two layers, allowing for more profound and complex feature extraction from the input image.
2. Filter size: A small filter size (3x3) is selected to focus on small details in the image.
3. Number of Filters: Filters increase with each block to capture more complex features.

Max Pooling

Max Pooling is used after each convolution block to reduce spatial dimensions and the number of parameters and avoid overfitting. This also helps introduce more stable features.

Activation (ReLU)

ReLU is used after each convolution layer to overcome the vanishing gradient problem and speed up convergence during training.

Flatten

Converts the output of the convolution layer to a 1D vector for input to the fully connected layer.

Dense Layers with Dropout

Dense layers with dropouts prevent overfitting and ensure the model learns more general features. Dropout randomly ignores some units during training to improve generalization.

Output Layer with Softmax

The output layer uses softmax to generate a probability distribution over possible classes, allowing unambiguous classification.

Experiment Set Up with AlexNet

Figure 7 shows the architecture of AlexNet. A ReLU dogs five convolution layers. Each convolution layer has a different kernel filter, illustrating extracting features from the given image. After each convolution layer, there are three max-pooling layers whose function is to reduce the image dimensions by taking the maximum value within the window.

```
Model: "model"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_1 (InputLayer)        [(None, 224, 224, 3)]     0

 conv2d (Conv2D)             (None, 222, 222, 16)      448

 conv2d_1 (Conv2D)           (None, 220, 220, 32)      4640

 activation (Activation)     (None, 220, 220, 32)      0

 max_pooling2d (MaxPooling2D  (None, 110, 110, 32)     0
 )

 conv2d_2 (Conv2D)           (None, 108, 108, 32)      9248

 conv2d_3 (Conv2D)           (None, 106, 106, 32)      9248

 activation_1 (Activation)   (None, 106, 106, 32)      0

 max_pooling2d_1 (MaxPooling  (None, 53, 53, 32)       0
 2D)

 conv2d_4 (Conv2D)           (None, 51, 51, 64)        18496

 conv2d_5 (Conv2D)           (None, 49, 49, 64)        36928

 activation_2 (Activation)   (None, 49, 49, 64)        0

 max_pooling2d_2 (MaxPooling  (None, 24, 24, 64)       0
 2D)

 flatten (Flatten)           (None, 36864)             0

 dense (Dense)               (None, 64)                2359360

 dropout (Dropout)           (None, 64)                0

 dense_1 (Dense)             (None, 3)                 195

=================================================================
Total params: 2,438,563
Trainable params: 2,438,563
Non-trainable params: 0
_________________________________________________________________
```

Figure 6. Proposed CNN Architecture

After the last convolution layer, the output of that layer is flattened with size (256), and three dense layers are dogged by a dropout layer. Dropout layers reduce overfitting by randomly turning off some nodes during training. The dense layer is responsible for fully connecting each node of the previous layer. The final layer uses a softmax activation function to generate output class probabilities. This AlexNet model is designed to perform multi-class classification with three output classes.

The total number of parameters in this AlexNet model is 28,040,483, including trainable weight parameters. Because they are all set as trainable parameters, all of these parameters can be trained during the training process.

Reasoning and explanation for each layer in AlexNet architecture:

1. Filter size and stride: Different filter sizes and strides are used to capture different levels of detail in the image. For example, an 11x11 filter with a 4x4 stride captures coarse features in the initial image, while a 3x3 filter with a 1x1 stride captures finer details in deeper layers.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 54, 54, 96)        34944

 max_pooling2d (MaxPooling2D  (None, 27, 27, 96)       0
 )

 conv2d_1 (Conv2D)           (None, 17, 17, 256)       2973952

 max_pooling2d_1 (MaxPooling  (None, 8, 8, 256)        0
 2D)

 conv2d_2 (Conv2D)           (None, 6, 6, 384)         885120

 conv2d_3 (Conv2D)           (None, 4, 4, 384)         1327488

 conv2d_4 (Conv2D)           (None, 2, 2, 256)         884992

 max_pooling2d_2 (MaxPooling  (None, 1, 1, 256)        0
 2D)

 flatten (Flatten)           (None, 256)               0

 dense (Dense)               (None, 4096)              1052672

 dropout (Dropout)           (None, 4096)              0

 dense_1 (Dense)             (None, 4096)              16781312

 dropout_1 (Dropout)         (None, 4096)              0

 dense_2 (Dense)             (None, 1000)              4097000

 dropout_2 (Dropout)         (None, 1000)              0

 dense_3 (Dense)             (None, 3)                 3003

=================================================================
Total params: 28,040,483
Trainable params: 28,040,483
Non-trainable params: 0
_________________________________________________________________
```

Figure 7. Proposed AlexNet Architecture

2. Activation Functions: ReLU is used after each convolution layer to speed up convergence and overcome the vanishing gradient problem.
3. MaxPooling2D Layers: Pool size and stride: Pooling reduces an image's spatial dimensions, reducing the number of parameters and computations required in subsequent layers and helping prevent overfitting.
4. Flatten Layer: This function converts the 3D output of the last convolution layer to a 1D vector for input to the fully connected layer.
5. Dense Layers with Dropout: The dropout technique prevents overfitting by randomly ignoring some units during training.
6. Activation: ReLU is used in the first two dense layers to exploit their nonlinearity, and softmax is used in the last layer to generate a probability distribution over possible classes.
7. Output Layer (Softmax): Used to generate the probability of each output class, which facilitates multi-class classification.

Result

Table 4 compares two model architectures: the proposed CNN Architecture and the AlexNet

Architecture, which is based on evaluation metrics carried out at the validation and testing stages.

AlexNet Architecture performs better than CNN Architecture in terms of loss and accuracy in both the validation and testing stages. Compared to CNN Architecture, AlexNet has a lower validation loss (0.0261 vs. 0.0377 (Figure 8)) and higher validation accuracy (0.9962 vs. 0.9925 (Figure 9)), likewise, for testing loss (0.0646 vs. 0.0572 (Figure 10)) and testing accuracy (0.9900 vs. 0.9800 (Figure 11)).

Both model architectures show excellent results in terms of recall, precision, and F-1 score. However, the AlexNet Architecture is slightly superior, with Recall, Precision, and F-1 values reaching 0.99, while the Proposed CNN Architecture has a value of 0.98. This shows that the AlexNet Architecture has a slightly better ability to classify data correctly.

Although both show excellent results, the AlexNet Architecture tends to perform better than the Proposed CNN Architecture.

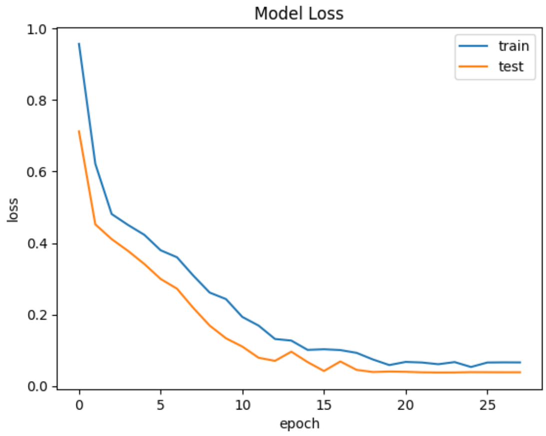

Figure 8. The Loss of CNN Architecture

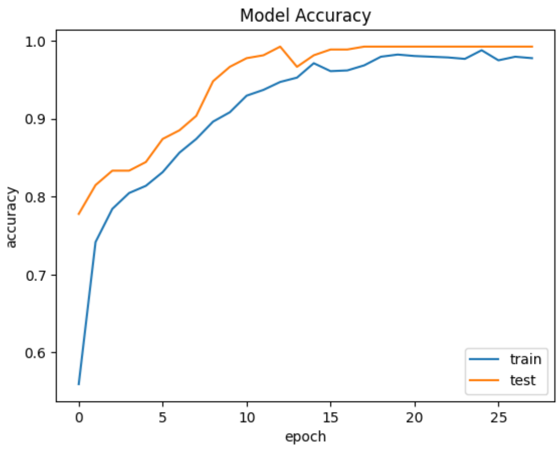

Figure 9. The Accuracy of CNN Architecture

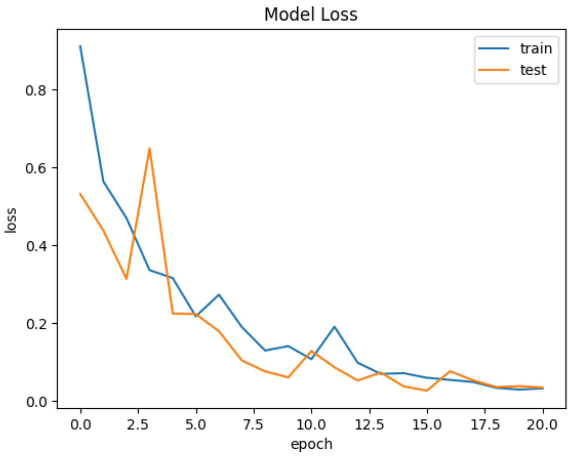

Figure 10. The Loss of AlexNet Architecture

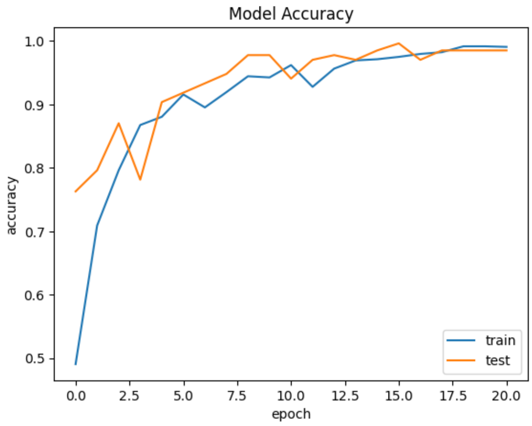

Figure 11. The Accuracy of AlexNet Architecture

The use of early stopping in the training of the CNN and AlexNet models shows that although the maximum number of epochs is set to be the same (50 epochs), training is stopped early (at the 28th epoch for CNN and the 21st epoch for AlexNet) to prevent overfitting. This shows that each model has a different optimal convergence point based on the interaction between the model architecture and the training data.

Figure 12 shows the confusion matrix for CNN architecture. Class 0 (rotten palm fruit): Of the 50 samples, 49 were correctly classified as Class 0 and only 1 was incorrectly classified as Class 2. Class 1 (ripe palm oil fruit): Of the 50 samples, 48 were correctly classified as Class 1 and 2 were incorrectly classified as Class 2. Class 2 (Raw palm oil fruit): All samples (50) were correctly classified as Class 2.

Figure 13 shows the confusion matrix for AlexNet. Class 0 (rotten palm oil fruit): Of the 50 samples, 49 were correctly classified as Class 0, and only one was misclassified as Class 2.

Table 4. Comparison of Both Architecture

Architecture	Validation		Testing		Precision	Recall	F-1 Score
	Loss	Acc	Loss	Acc			
Proposed CNN Architecture	0.0377	0.9925	0.0572	0.9800	0.98	0.98	0.98
AlexNet Architecture	0.0261	0.9962	0.0646	0.9900	0.99	0.99	0.99

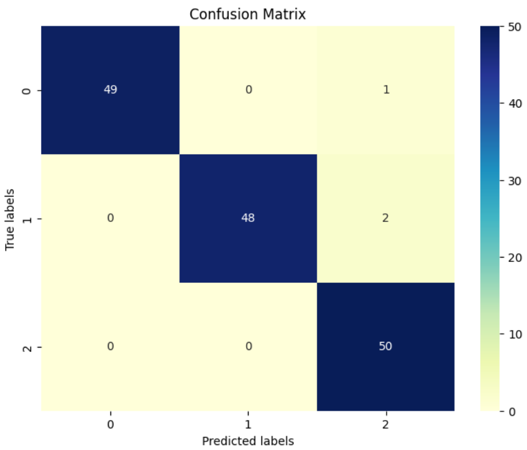

Figure 12. Confusion Matrix of CNN Architecture

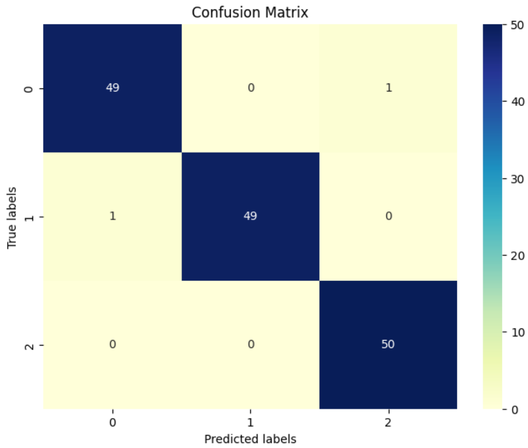

Figure 13. Confusion Matrix of AlexNet Architecture

Class 1 (ripe palm oil fruit): Of the 50 samples, 49 were correctly classified as Class 1, and 1 was incorrectly classified as Class 0. Class 2 (raw palm oil fruits): All samples (50) were correctly classified as Class 2.

Figure 14 shows the classification results of 100 test data on the CNN architecture. Out of the 100 tested data, there were 3 image data misclassifications. Figure 15 shows the classification results of 100 test data on the AlexNet architecture. It can be seen that out of the 100 tested data, there were 2 image data misclassifications. Comparison performance for both models:

1. Classification performance: Both models performed very well in classifying class 2 (raw palm oil fruit), with all samples classified correctly.
2. Classes with classification errors: Both models have slight errors in classifying class 0 (rotten palm oil fruit) and class 1 (ripe palm oil fruit). However, the CNN model has more errors in classifying class 1, while the AlexNet model has more errors in class 0.
3. Consistency: The AlexNet model performs more consistently in classifying class 0 and class 1, with fewer errors compared to the CNN model.

Discussion

The data analysis compares the performance of two deep learning architectures in classifying the ripeness of palm oil. The results highlight exciting discussions that can be drawn from these experiments, including:

1. Model Architecture Comparison: The data shows that AlexNet consistently performs better than the CNN architecture in terms of loss, accuracy, recall, precision, and F-1 Score. AlexNet is very suitable for palm oil fruit ripeness classification.
2. Model complexity vs. performance: The AlexNet architecture has more layers and parameters than the proposed CNN architecture. This case may raise the question of how important model complexity is in achieving better performance and the extent to which simpler models can compete with more complex models in the classification context.
3. AlexNet vs. custom CNN: This comparison also raises questions about the benefits and limitations of using transfer learning (as in AlexNet) versus building custom models (as in the proposed CNN architecture) in the context of classification applications. This answers whether AlexNet provides better results than the CNN model regarding palm fruit classification.

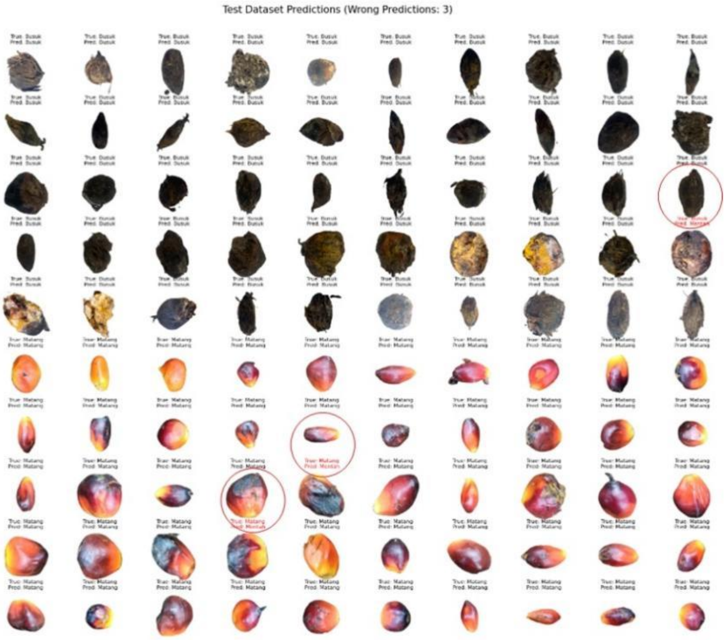

Figure 14. Testing with 100 data tests with CNN Architecture

Figure 15. Testing with 100 data tests with AlexNet

Based on previous research that employed various approaches to classify the maturity level of oil palm fruit, the results varied depending on the methods and algorithms used. For instance, a study [16] used a contour-based approach combined with the Canny algorithm for segmenting oil palm fruit based on shape and color features, achieving 90.13% accuracy. Meanwhile, a study [17] combined color- and texture-based feature extraction with feature selection using PCA, reaching an accuracy of 98.3% with an Artificial Neural Network (ANN). Other studies, such as [18] and [19], employed image processing techniques based on color characteristics and the BGLAM+ANN algorithm for maturity classification, with 94% and 93% accuracy, respectively.

However, newer approaches, such as the one proposed by [20] using YOLOv5 object detection algorithms, achieved only 84% accuracy. In this context, the classification of oil palm fruit maturity using the AlexNet architecture demonstrated a significant improvement, achieving an accuracy of 99%. This remarkable result was achieved through optimized preprocessing methods and careful hyperparameter tuning, surpassing previous studies with various other methods. Therefore, AlexNet proves to be more effective for handling oil palm fruit maturity classification, especially when combined with precise image processing strategies and model adjustments.

CONCLUSION

AlexNet is a significant landmark in developing CNN and computer image processing, especially in palm oil fruit ripeness classification. By introducing a deep architecture, ReLU activation functions, dropout techniques, and GPUs for fast training, AlexNet managed to overcome existing problems and achieve superior performance in image recognition. With its success, AlexNet paved the way for further advances in computer vision, including object recognition, image segmentation, and natural language processing. AlexNet's architecture, techniques, and contributions remain the foundation for many modern CNN models used in various applications, including facial recognition, autonomous cars, and medical recognition. AlexNet is a successful CNN model for image classification tasks and the starting point for a revolution in computer vision and deep learning in general. Further research can be done to explore more sophisticated and adaptive transfer learning strategies such as VGGNet, DenseNet, EfficientNet, ResNet, and Inception.

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 the Department of Computer System Engineering mentors for their invaluable guidance and feedback throughout the research process. We also 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.

DATA AVAILABILITY

All data can be accessed by emailing the corresponding author with a valid reason for the request.

REFERENCES

[1] H. Herman, A. Susanto, T. W. Cenggoro, S. Suharjito, and B. Pardamean, "Oil Palm Fruit Image Ripeness Classification with Computer Vision using Deep Learning and Visual Attention," *J. Telecommun. Electron. Comput. Eng.*, vol. 12, no. 2, pp. 21–27, 2020,

[2] 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 (Switzerland)*, vol. 12, no. 10, pp. 14179–14195, 2012, doi: 10.3390/s121014179.

[3] 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 Conf. Ser. Earth Environ. Sci.*, vol. 20, no. 1, 2014, doi: 10.1088/1755-1315/20/1/012017.

[4] A. Y. Saleh and E. Liansitim, "Palm oil classification using deep learning," *Sci. Inf. Technol. Lett.*, vol. 1, no. 1, pp. 1–8, 2020, doi: 10.31763/sitech.v1i1.1.

[5] Suharjito *et al.*, "Annotated Datasets of Oil Palm Fruit Bunch Piles for Ripeness Grading Using Deep Learning," *Sci. Data*, vol. 10, no. 1, pp. 1–9, 2023, doi: 10.1038/s41597-023-01958-x.

[6] Z. Y. Wong, W. J. Chew, and S. K. Phang, "Computer vision algorithm development for classification of palm fruit ripeness," in *AIP Conference Proceedings*, 2020. doi: 10.1063/5.0002188.

[7] 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 *5th International Conference on Informatics and Computing, ICIC*, 2020. doi: 10.1109/ICIC50835.2020.9288603.

[8] N. Tajbakhsh *et al.*, "Convolutional Neural Networks for Medical Image Analysis: Full Training or Fine Tuning?," *IEEE Trans. Med. Imaging*, vol. 35, no. 5, pp. 1299–1312, 2016, doi: 10.1109/TMI.2016.2535302.

[9] D. Anand, O. I. Khalaf, F. Hajjej, W. Wong, S. Pan, and G. R. Chandra, "Optimized Swarm Enabled Deep learning technique for bone tumor detection using Histopathological Image," *SINERGI*, vol. 27, no. 3, pp. 451–466, 2023, doi: 10.22441/sinergi.2023.3.016.

[10] A. Manandhar, L. Hoegner, and U. Stilla, "Palm Tree Detection Using Circular Autocorrelation of Polar Shape Matrix," *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.*, vol. III–3, no. July, pp. 465–472, 2016, doi: 10.5194/isprsannals-iii-3-465-2016.

[11] M. Culman, S. Delalieux, and K. Van Tricht, "Individual palm tree detection using deep learning on RGB imagery to support tree

inventory," *Remote Sens.*, vol. 12, no. 21, pp. 1–31, 2020, doi: 10.3390/rs12213476.

[12] K. Yarak, A. Witayangkurn, K. Kritiyutanont, C. Arunplod, and R. Shibasaki, "Oil palm tree detection and health classification on high-resolution imagery using deep learning," *Agric.*, vol. 11, no. 2, pp. 1–17, 2021, doi: 10.3390/agriculture11020183.

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

[14] T. Shanthi and R. S. Sabeenian, "Modified Alexnet architecture for classification of diabetic retinopathy images," *Comput. Electr. Eng.*, vol. 76, pp. 56–64, 2019, doi: 10.1016/j.compeleceng.2019.03.004.

[15] E. L. Omonigho, M. David, A. Adejo, and S. Aliyu, "Breast Cancer:Tumor Detection in Mammogram Images Using Modified AlexNet Deep Convolution Neural Network," in *International Conference in Mathematics, Computer Engineering and Computer Science, ICMCECS*, 2020. doi: 10.1109/ICMCECS47690.2020.240870.

[16] A. Septiarini, H. Hamdani, H. R. Hatta, and K. Anwar, "Automatic image segmentation of oil palm fruits by applying the contour-based approach," *Sci. Hortic. (Amsterdam).*, vol. 261, no. October, p. 108939, 2020, doi: 10.1016/j.scienta.2019.108939.

[17] 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," *Sci. Hortic. (Amsterdam).*, vol. 286, no. October 2020, p. 110245, 2021, doi: 10.1016/j.scienta.2021.110245.

[18] M. S. M. Alfatni, A. R. Mohamed Shariff, O. M. Ben Saaed, A. M. Albhbah, and A. Mustapha, "Colour Feature Extraction Techniques for Real Time System of Oil Palm Fresh Fruit Bunch Maturity Grading," *IOP Conf. Ser. Earth Environ. Sci.*, vol. 540, no. 1, 2020, doi: 10.1088/1755-1315/540/1/012092.

[19] 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," *Agric.*, vol. 12, no. 9, 2022, doi: 10.3390/agriculture12091461.

[20] M. Y. M. A. Mansour, K. D. Dambul, and K. Y. Choo, "Object Detection Algorithms for Ripeness Classification of Oil Palm Fresh Fruit Bunch," *Int. J. Technol.*, vol. 13, no. 6, pp. 1326–1335, 2022, doi: 10.14716/ijtech.v13i6.5932.

[21] A. H. Pratomo, W. Kaswidjanti, A. S. Nugroho, and S. Saifullah, "Parking detection system using background subtraction and hsv color segmentation," *Bull. Electr. Eng. Informatics*, vol. 10, no. 6, pp. 3211–3219, 2021, doi: 10.11591/eei.v10i6.3251.

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

[23] Y. Sun, B. Xue, M. Zhang, and G. G. Yen, "Evolving Deep Convolutional Neural Networks for Image Classification," *IEEE Trans. Evol. Comput.*, vol. 24, no. 2, pp. 394–407, 2020, doi: 10.1109/TEVC.2019.2916183.

[24] S. Albawi, T. A. M. Mohammed, and S. Alzawi, "Understanding of a Convolutional Neural Network," in *Icet2017*, 2017, pp. 1–6. doi: 10.1109/ICEngTechnol.2017.8308186.

[25] A. Nasiri, A. Taheri-Garavand, and Y. D. Zhang, "Image-based deep learning automated sorting of date fruit," *Postharvest Biol. Technol.*, vol. 153, no. January, pp. 133–141, 2019, doi: 10.1016/j.postharvbio.2019.04.003.

[26] H. Jiang and E. Learned-Miller, "Face Detection with the Faster R-CNN," in *12th IEEE International Conference on Automatic Face and Gesture Recognition*, 2017, pp. 650–657. doi: 10.1109/FG.2017.82.

[27] A. W. Salehi *et al.*, "A Study of CNN and Transfer Learning in Medical Imaging: Advantages, Challenges, Future Scope," *Sustain.*, vol. 15, no. 7, 2023, doi: 10.3390/su15075930.

[28] A. Kamilaris and F. X. Prenafeta-Boldú, "Deep learning in agriculture: A survey," *Comput. Electron. Agric.*, vol. 147, no. July 2017, pp. 70–90, 2018, doi: 10.1016/j.compag.2018.02.016.

[29] S. Y. Sen and N. Ozkurt, "Convolutional Neural Network Hyperparameter Tuning with Adam Optimizer for ECG Classification," in *Innovations in Intelligent Systems and Applications Conference, ASYU*, 2020. doi: 10.1109/ASYU50717.2020.9259896.

[30] A. Gupta, R. Ramanath, J. Shi, and S. S. Keerthi, "Adam vs. SGD: Closing the generalization gap on image classification," in *OPT2021: 13th Annual Workshop on Optimization for Machine Learning*, 2021,

pp. 1–7.
[31] D. Soydaner, D. Soydaner, and A. In-, “A Comparison of Optimization Algorithm for Deep LEarning,” *Int. J. Pattern Recognit. Artif. Intelli- gence*, vol. 34, no. 13, 2020, doi: 10.1142/S0218001420520138.
[32] A. A. Almisreb, N. Jamil, and N. M. Din, “Utilizing AlexNet Deep Transfer Learning for Ear Recognition,” in *4th International Conference on Information Retrieval and Knowledge Management: Diving into Data Sciences, CAMP*, IEEE, 2018, pp. 8–12. doi: 10.1109/INFRKM.2018.8464769.
[33] 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,” *Int. J. Appl. Math. Comput. Sci.*, vol. 31, no. 4, pp. 549–561, 2021, doi: 10.34768/amcs-2021-0037.
[34] Susanto, D. Stiawan, M. A. S. Arifin, M. Y. Idris, and R. Budiarto, “Effective and efficient approach in IoT Botnet detection,” *SINERGI*, vol. 28, no. 1, pp. 31–42, 2024, doi: 10.22441/sinergi.2024.1.004.
[35] A. Ashraf et al.," Detection of Road Cracks Using Convolutional Neural Networks and Threshold Segmentation," *Journal of Integrated and Advanced Engineering (JIAE)*, vol. 2, no. 2, pp. 123-134, 2022, doi: 10.51662/jiae.v2i2.82