

Article

Computationally Efficient Transfer Learning Pipeline for Oil Palm Fresh Fruit Bunch Defect Detection

Yang Luo ¹, Anwar P. P. Abdul Majeed ^{2,*}, Zaid Omar ³, Saad Aslam ² and Yi Chen ^{1,*}¹ School of Intelligent Manufacturing Ecosystem, Xi'an Jiaotong-Liverpool University, Taicang, Suzhou 215400, China; yang.luo@xjtlu.edu.cn² School of Computing and Artificial Intelligence, Faculty of Engineering and Technology, Sunway University, Bandar Sunway, Subang Jaya 47500, Malaysia; saada@sunway.edu.my³ Faculty of Electrical Engineering, Universiti Teknologi Malaysia, Johor Bahru 81310, Malaysia; zaidomar@utm.my

* Correspondence: anwarm@sunway.edu.my (A.P.P.A.M.); yi.chen02@xjtlu.edu.cn (Y.C.)

Abstract: The present study addresses the inefficiencies of the manual classification of oil palm fresh fruit bunches (FFBs) by introducing a computationally efficient alternative to traditional deep learning approaches that require extensive retraining and large datasets. Using feature-based transfer learning, where pre-trained Convolutional Neural Network architectures, namely EfficientNet_B0, EfficientNet_B4, ResNet152, and VGG16, serve as fixed feature extractors coupled with the Logistic Regression classifier, this research evaluated the performance on a dataset of 466 images categorized as defective or non-defective. The results demonstrate a robust classification performance across all architectures, with the EfficientNet_B4-LR pipeline achieving an exceptional accuracy value of 96.81%, which was further enhanced through hyperparameter optimization. This confirms that feature-based transfer learning offers a reliable, resource-efficient, and practical solution for automated FFB defect detection that can significantly benefit the palm oil industry by providing a scalable alternative to subjective manual-grading methods.

Keywords: fresh fruit bunch; defect detection; transfer learning; feature extraction; deep learning



Academic Editor: Juan Gabriel Avina-Cervantes

Received: 14 April 2025

Revised: 30 May 2025

Accepted: 4 June 2025

Published: 6 June 2025

Citation: Luo, Y.; P. P. Abdul Majeed, A.; Omar, Z.; Aslam, S.; Chen, Y. Computationally Efficient Transfer Learning Pipeline for Oil Palm Fresh Fruit Bunch Defect Detection.

Technologies **2025**, *13*, 234.
<https://doi.org/10.3390/technologies13060234>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The global palm oil industry represents a significant economic sector, with a market value exceeding USD 50 billion in 2021 [1]. Projections at that time indicated expansion at a compound annual growth rate (CAGR) of at least 4%, potentially reaching USD 65 billion by 2027 [1]. This expansion highlights palm oil's integral role across diverse industries and its importance in supporting the livelihoods of millions globally, particularly within major producing nations, such as Indonesia and Malaysia, where it employs nearly 5 million individuals directly and supports millions more indirectly [2]. Nonetheless, the quality assessment of fresh fruit bunches (FFBs) remains critical for optimizing oil yield and maintaining industry standards, yet traditional manual inspection methods are labor-intensive and prone to subjective inconsistencies.

Recent advances in machine learning, particularly, Convolutional Neural Networks (CNNs), offer promising solutions for automated FFB classification, primarily due to their effectiveness in image analysis tasks. Nonetheless, it is worth noting at this juncture that developing a CNN model from scratch is computationally expensive. To address the aforesaid quagmire, an alternative approach is presented, which utilizes pre-trained CNNs

as fixed feature extractors coupled with simpler classifiers, potentially offering enhanced computational efficiency and robust performance with limited data. This approach is known as feature-based transfer learning.

Despite the extensive research on FFB ripeness classification, the critical task of distinguishing defective bunches (rotten, diseased, damaged, and empty) from non-defective ones across various maturity stages remains underexplored using feature-based transfer learning. This approach could provide significant advantages, including reduced training complexity, lower computational costs, and improved generalization, when labeled defect data are limited.

This study systematically evaluates feature-based transfer learning pipelines combining prominent pre-trained CNN architectures (EfficientNet-B0, EfficientNet-B4, ResNet152, and VGG16) as feature extractors with Logistic Regression classifiers for automated FFB defect detection. Our comprehensive analysis examines performance under varying hyperparameter conditions to identify optimal configurations and demonstrate practical viability for industry deployment.

The remainder of this paper is organized as follows: Section 2 provides a review of the related literature on FFB classification, further defining the existing research gaps. Section 3 details the materials and methodologies employed in this study. Section 4 presents and discusses the experimental results obtained. Finally, Section 5 offers conclusions, discusses the implications of the findings, and provides recommendations for future research directions.

2. Related Works

Automated FFB analysis has evolved significantly from traditional manual inspection toward machine learning solutions. This section reviews the relevant approaches across three key paradigms, namely fine-tuning deep networks, object detection frameworks, and alternative feature engineering methods, highlighting the research gap our study addresses. While much research has addressed the classification of FFBs based on ripeness levels, the equally critical task of distinguishing defective bunches (e.g., rotten, diseased, damaged, and empty) from non-defective ones across various maturity stages presents distinct challenges and remains essential for comprehensive quality control.

Within the domain of automated FFB analysis, fine-tuning pre-trained networks have dominated the FFB classification research. Foundational work by Ibrahim et al. [3] highlighted the significant advantages of fine-tuning a pre-trained AlexNet model over training a CNN from scratch or using traditional Support Vector Machines (SVMs) with hand-crafted features, achieving high accuracy for ripeness classification on their dataset. This approach's efficacy was further corroborated by subsequent studies employing various CNN architectures. For example, Kurniawan et al. [4] achieved over 93% accuracy in classifying FFBs as “raw”, “ripe”, or “rotten” by fine-tuning AlexNet. Their later research [5] extended this success to deeper architectures, like ResNet-152V2 and DenseNet-201, again demonstrating exceptional accuracies via fine-tuning for ripeness classification.

Deep learning-based object detection frameworks, notably the YOLO (You Only Look Once) family, which facilitate simultaneous FFB localization and classification, have also been investigated. Similarly, Mansour et al. [6], after evaluating several models, identified fine-tuned YOLOv5m as highly effective for ripeness classification, achieving 84.2% mean Average Precision (mAP). Object detection models have proven to be adaptable to more complex grading scenarios as well. Junior et al. [7] utilized fine-tuned YOLOv4 variants for real-time video analysis, successfully classifying FFBs into six categories, including “empty” and “abnormal” bunches, reaching up to 90.6% mAP with a lightweight model. Furthermore, Suhajito et al. [8] developed an end-to-end system deploying a fine-tuned, quantized YOLOv4 on smartphones for practical mill-site grading, explicitly addressing defective cate-

gories and achieving a notable validation performance (~99% mAP). Recognizing the often data-intensive nature of deep learning, some research explores alternative pathways better suited to scenarios with limited datasets. For instance, Rosbi et al. [9] proposed an outdoor FFB classification system that integrated sophisticated pre-processing with traditional feature extraction. Their method involved localizing FFBs using YOLOv4-Tiny, segmenting the background via superpixel-based Fast Fuzzy C-means (FFCM) clustering, and subsequently extracting hand-crafted features, such as color moments and opposite-color local binary patterns from the segmented FFB regions. These engineered features were then classified using multilayer perceptron to predict five classes, including “damaged” and “empty” bunches. Achieving 93.68% accuracy, this approach demonstrated the potential viability of combining advanced segmentation and engineered features, particularly in contexts where end-to-end deep learning fine-tuning might be challenging due to data constraints.

Despite such explorations involving traditional feature engineering, an alternative deep learning transfer strategy, feature-based transfer learning appears to be investigated less within the FFB classification context. This technique utilizes a pre-trained deep network, typically a CNN, as a fixed feature extractor. Activations from intermediate layers serve as feature vectors for training a separate, often simpler, machine learning classifier (e.g., SVM and LR). The related work, for example, Suhajito et al.’s study [10], employed lightweight CNNs focusing on efficient feature representation, and Ibrahim et al. [3] contrasted deep learning against SVMs using hand-crafted features, where the explicit, systematic evaluation of using features extracted from deep pre-trained networks with separate classifiers specifically for FFB defect detection, seems limited. Mamat et al. [11], while effectively using YOLO’s pre-trained knowledge for annotation, did not focus on implementing a feature-extraction classification pipeline.

While fine-tuning approaches dominate the current research, feature-based transfer learning by using pre-trained CNNs as fixed feature extractors with separate classifiers remains systematically underexplored for FFB defect detection. This gap is significant as feature-based approaches offer several advantages over fine-tuning. These include significantly reduced training complexity and computational costs, as only the final classifier requires training. Furthermore, it may offer an improved generalization performance when labeled data for specific defect types is limited and provides flexibility in combining features from different network layers or models to capture diverse visual characteristics relevant to defects. Therefore, the present study aims at addressing this gap by systematically evaluating feature-based transfer learning pipelines across diverse CNN architectures, providing crucial insights into computational efficiency and performance trade-offs for practical FFB defect detection systems.

3. Materials and Methods

This section details the methodology employed to develop and evaluate the proposed system for classifying FFBs into defective or non-defective categories. The subsequent subsections describe the key components of this study, beginning with the assembly and characteristics of the dataset used (Section 3.1). Following this, the rationale for adopting a transfer learning approach is discussed (Section 3.2), along with the specific methods for selecting and utilizing CNN architectures as feature extractors (Section 3.3). The proposed experimental framework combining these elements is then outlined (Section 3.4). Finally, the specifics of the LR classifier employed (Section 3.5) and the metrics used for the performance evaluation (Section 3.6) are presented.

3.1. Dataset Description

A collection of 466 oil palm FFB images [12] was gathered from plantations across Johor, Negeri Sembilan, and Perak, Malaysia. Initially, an experienced grader assigned ripeness labels to each image based on a five-stage categorization. However, for the purpose of this proof-of-concept study focusing on defect detection, these labels were consolidated into two primary categories, viz. defective and non-defective. This re-categorization paid particular attention to visual indicators of suboptimal quality, such as the presence of empty sockets. To ensure the dataset mirrored realistic field scenarios, images were captured under diverse illumination conditions and from various angles, including FFBs located both on the tree and on the ground post-harvest.

The resulting dataset encompasses complex visual conditions, where factors like uneven lighting and background clutter were present. While these conditions increase the difficulty of the classification task, they also enhance the practical relevance of this study's findings. Figure 1 provides illustrative examples differentiating between defective and non-defective FFBs, highlighting key visual cues, such as external coloration, socket emptiness, and fruitlet detachment. It is worth noting that in the present study, no data augmentation is performed as this study aims to illustrate that, with a limited dataset, the pipeline would still be able to perform well.



Figure 1. Sample images of oil palm fresh fruit bunches (FFBs) from the dataset, illustrating (a) a defective bunch and (b) a non-defective bunch.

3.2. Feature Extraction: Transfer Learning

Transfer learning is a deep learning methodology, where a model pre-trained on a large source task is adapted to solve a different yet related target task [13]. This technique allows leveraging knowledge encapsulated within models trained on extensive datasets, such as ImageNet [14], for application to tasks with comparatively smaller datasets, like the FFB classification addressed in this study. Figure 2 illustrates the general transfer learning framework employed here. The core idea involves utilizing a complex, pre-trained model as a foundation, thereby transferring its learned feature representations to the new target task. Instead of initiating training with randomly initialized weights, transfer learning exploits the inherent hierarchical feature learning capabilities of deep neural networks [15].

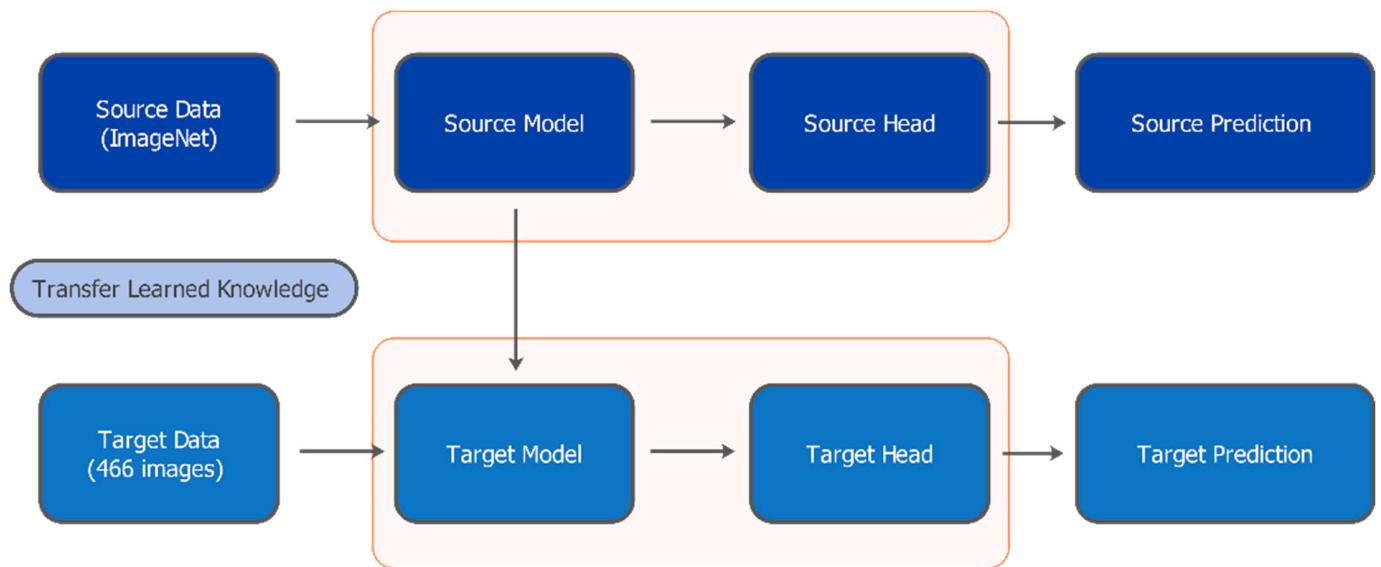


Figure 2. The transfer learning framework employed in this study for FFB classification.

Within this framework, the initial layers of the pre-trained network typically function as generalized feature extractors, adept at capturing low-level visual primitives, like edges, colors, basic textures, and gradients. Subsequent middle layers learn to detect more complex patterns and object parts by combining these lower-level features. Finally, the deeper layers tend to learn higher-level, potentially more task-specific feature combinations. This hierarchical structure makes transfer learning particularly effective when the source and target domains share similar low-level visual characteristics. Moreover, the adaptation to the new task often primarily involves modifying or retraining only the final layers responsible for the classification, while keeping the previous feature extraction layers largely unchanged or only fine-tuned.

Employing transfer learning offers several significant advantages. It often reduces the need for extensive hyperparameter tuning specific to the network architecture, streamlining the model development workflow. Furthermore, it considerably lowers the computational resources and time investment typically required for training deep models from scratch by reusing the optimized weights learned during the extensive pre-training phase. Crucially, transfer learning can also serve as an effective regularization technique, helping to mitigate the risk of overfitting, which is a common challenge when training complex models on limited labeled data [16]. Owing to these benefits, this technique has seen widespread adoption across diverse computer vision applications, including medical image analysis [17,18], agricultural monitoring [19], industrial product quality control [20], and smart manufacturing [21]. Its capacity to facilitate the development of accurate models even with constrained data resources makes it invaluable in specialized domains, where acquiring large-scale labeled datasets is often impractical or prohibitively expensive.

3.3. Pre-Trained Models

For feature extraction in this study, we utilized the pre-trained CNN architectures as fixed feature extractors without modifying their weights. Features were extracted from the final convolutional layer output (global average pooling layer) of each pre-trained CNN, before the original classification head. This approach yields feature vectors of specific dimensions for each architecture, i.e., 1280 features for EfficientNet-B0, 1792 features for EfficientNet-B4, 2048 features for ResNet152, and 512 features for VGG16. These high-dimensional feature representations encapsulate the hierarchical visual patterns learned

during pre-training on ImageNet, providing rich semantic information for the subsequent classification by the Logistic Regression classifier.

After assembling the dataset and outlining a transfer learning strategy, the next step was to determine which pre-trained models to adopt. The chosen architectures, EfficientNet-B0, EfficientNet-B4, ResNet152, and VGG16, span a progression from an ultra-light to a heavyweight design. This spectrum provides a balanced exploration of different parameter complexities and design philosophies, allowing for a more comprehensive assessment of how each model handles FFB images. The subsequent subsections describe the technical aspects and motivation behind each selection.

3.3.1. EfficientNet-B0 Architecture (Ultra-Lightweight)

EfficientNet-B0, introduced by Tan and Le [22], offers a carefully balanced compound-scaling strategy, which coordinates depth, width, and resolution in a unified manner. This baseline variant uses a family of Mobile Inverted Bottleneck Convolution (MBConv) blocks coupled with Squeeze-and-Excitation (SE) modules. The MBConv blocks employ depthwise separable convolutions, which cut down on computational complexity by dividing standard convolutions into channel-wise and pointwise operations. The SE components dynamically recalibrate feature channels according to the global context, often improving the representational efficiency for each feature map.

The original motivation for choosing EfficientNet-B0 is its relatively small number of parameters (5.3 million) despite the competitive accuracy. Its capacity-to-size ratio is especially appealing for tasks constrained by hardware resources, and its generalizable representations have demonstrated good results in agricultural imagery. By starting with B0, the project aimed to assess whether an ultra-lightweight model would suffice to distinguish defective from non-defective FFBs, given a smaller data scale and wide variations in real plantation scenes.

3.3.2. EfficientNet-B4 Architecture (Lightweight)

EfficientNet-B4 builds upon the same design concepts found in EfficientNet-B0 but employs a more aggressive scaling coefficient across network width, depth, and resolution [22]. This scaling substantially increases the model's representational capacity, allowing it to capture more nuanced visual details in the input images. In the context of the FFB assessment, small surface-level distinctions, such as fine texture variations or subtleties in color patterns, can be crucial for differentiating defective from non-defective FFBs. By deploying a deeper and wider pipeline with a higher input resolution, EfficientNet-B4 is expected to learn a richer hierarchy of visual features, although it requires more computational resources and memory (about 19 M parameters). Including B4 in this study thus enables an understanding of whether the performance gains from a bigger model justify the higher computational cost when dealing with FFB classification.

3.3.3. ResNet152 Architecture (Medium Weight)

ResNet152, developed by He et al. [23], integrates residual blocks with skip connections to improve vanishing gradients in very deep neural networks. Each block adopts a bottleneck design that reduces the dimensionality prior to convolution, striking a balance between network depth and computational feasibility. In practice, deeper networks often excel at recognizing subtle or high-level features, which may be essential for complex tasks. The skip connections help preserve gradient flow, making it easier to optimize a 152-layer network without succumbing to gradient decay. ResNet152 is well known for its competitive accuracy on large-scale benchmarks and for serving effectively as a general-purpose feature extractor. Using ResNet152 in this study ensures that a thoroughly deep architecture

is included, allowing an assessment of whether extremely deep models provide tangible benefits in FFB classification compared to shallower but more efficient networks.

3.3.4. VGG16 Architecture (Heavy Weight)

VGG16, introduced by Simonyan and Zisserman [24], represents a foundation in modern CNN design. Although it is considered large in terms of the parameter count (approximately 138 M) and has a correspondingly high computational footprint, VGG16 is often favored for its simplicity: the network relies on consecutive 3×3 convolutions and 2×2 pooling operations, repeated systematically to increase depth. This uniform approach has historically led to a strong transfer learning performance across a variety of image domains. Despite its age and size, VGG16 remains a useful reference point for evaluating whether newer, more parameter-efficient models, such as EfficientNet or ResNet, might offer an equivalent or superior performance on specialized tasks, like FFB classification.

Table 1 compares the key attributes of these four networks, including parameter counts, file sizes, depths, and approximate FLOPs. Together, they cover a spectrum of architectural choices, from lightweight to heavyweight, and from recent innovations to classic designs to ensure that the final evaluation provides insights into how architecture size and structure affect FFB classification accuracy.

Table 1. Specifications of the pre-trained CNN architectures used as feature extractors in this study.

Model	Category	Parameters	Size	Depth	GFLOPs	Input Size
EfficientNet-B0	Ultra-lightweight	5.3 M	20 MB	82	0.39	$[224 \times 224 \times 3]$
EfficientNet-B4	Lightweight	19 M	75 MB	97	4.2	$[380 \times 380 \times 3]$
ResNet152	Medium weight	60 M	230 MB	152	11.5	$[224 \times 224 \times 3]$
VGG16	Heavy weight	138 M	528 MB	16	15.5	$[224 \times 224 \times 3]$

3.4. Proposed Experimental Approach

The overall experimental plan is illustrated in Figure 3, which depicts the transfer learning process applied to FFB images for defective vs. non-defective classifications. Each image was fed into a pre-trained deep network to extract discriminative features, which were then flattened and subsequently passed to a chosen machine learning classifier. The figure shows both the training phase, where the classifier learns from the extracted features, and the testing phase, which uses the trained model to predict labels on unseen data.

To ensure a robust performance estimation, the dataset was divided according to a 60:20:20 hold-out cross-validation strategy. A total of 60% of the data was used for training, 20% was reserved for validation, and the remaining 20% was allocated for testing. The validation set provides an unbiased sample for model fine-tuning, guarding against overfitting and preventing the inadvertent reuse of training data for hyperparameter selection. The test set was used exclusively at the final evaluation stage, so that the reported metrics accurately reflected the classifier's generalization capability. Table 2 shows the results of this hold-out validation and the image distribution across these three subsets.

Table 2. Distribution of FFB images across dataset splits and classes.

Classes	Training Set	Validation Set	Test Set	Overall
Defective	159	53	53	265
Non-Defective	120	40	41	201

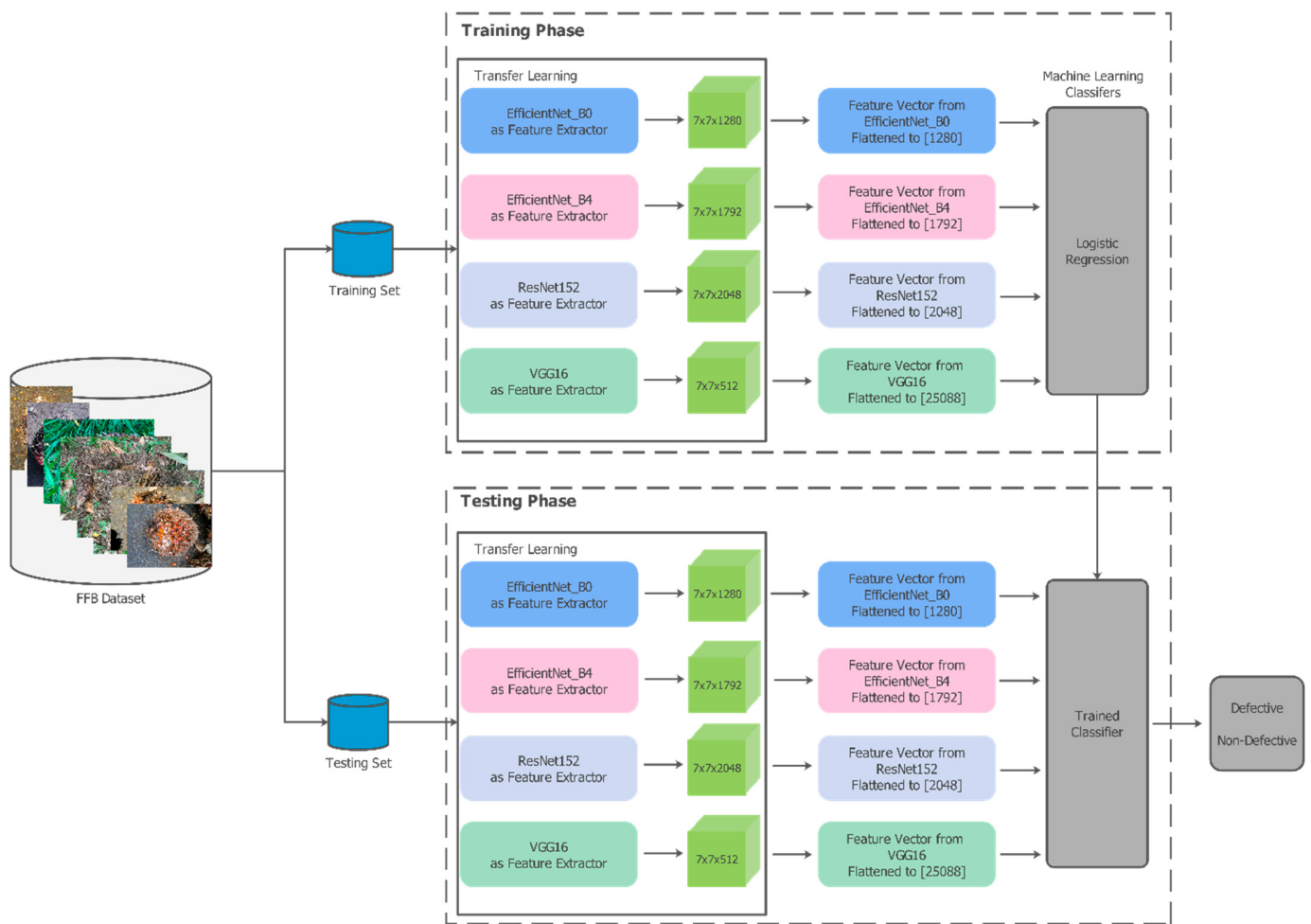


Figure 3. Proposed experimental framework for FFB classification using feature-based transfer learning.

3.5. Classifier: Logistic Regression

Logistic Regression (LR) is a widely used supervised machine learning approach for binary and multiclass classification tasks. By applying a sigmoid or logistic function, LR estimates class membership probabilities, making it particularly useful in domains where interpretability and simplicity are valued. In various applications, including healthcare analytics and financial prediction models, LR has proved itself to be both robust and computationally efficient.

In this study, we evaluated three primary hyperparameters for LR, namely (i) C , the inverse of the regularization strength that controls overfitting, (ii) the type of regularization penalty (L1 or L2), and (iii) the solver method (lbfgs, liblinear, or saga) as shown in Table 3. To optimize these hyperparameters, a grid-search-like procedure was implemented within the Optuna framework, using five-fold cross-validation on the training set to compute the classification accuracy for each candidate configuration. The best-performing set of hyperparameters was then retrained on the combined training and validation subsets, allowing the model to leverage all the available data for learning. This final model underwent a rigorous assessment with the test set, providing a conclusive measure of its ability to distinguish defective FFBs from their non-defective counterparts. By integrating both transfer learning for feature extraction and systematic LR hyperparameter tuning, this approach aimed to maximize the classification accuracy while retaining interpretability and computational feasibility.

Table 3. Hyperparameters investigated.

No	Hyperparameters	Tunable Values
1	Penalty	L1, L2, ElasticNet, None
2	Solver	lbfgs, liblinear, saga
3	C	0.001, 0.1, 1, 10, 1000

All experiments were performed on a high-performance computing platform equipped with an Intel Core i9-10940X CPU (3.30 GHz) and 64 GB of RAM, providing the resources needed for data-intensive operations and extensive model training. An NVIDIA GeForce RTX 4080 SUPER GPU, featuring 16 GB of dedicated memory and a total of 48 GB of available GPU memory, was used to accelerate deep learning computations. This robust hardware environment effectively managed large-scale datasets and complex model architectures, ensuring smooth and efficient processing. In conjunction with flexible libraries and frameworks, the system enabled a powerful transfer learning and classification pipeline optimized for high-accuracy outcomes on complex image data.

3.6. Performance Evaluation

The performance of the classifiers was evaluated using several standard metrics, including classification accuracy (CA), precision, recall, F1-score, and the confusion matrix. These metrics collectively offered a comprehensive evaluation of the classifier's predictive capability and reliability.

CA measures the overall success rate of the model, indicating how well it correctly classifies instances across all classes. It is computed as:

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

where true positive (TP) and true negative (TN) represent correctly predicted positive and negative instances, respectively. Conversely, false positive (FP) and false negative (FN) denote incorrect predictions for positive and negative instances.

Precision evaluates the proportion of true positive predictions among all positive predictions, quantifying the model's ability to avoid false positives. It is calculated as:

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

Recall, also known as sensitivity or true positive rate, measures the proportion of actual positive instances correctly identified by the model, highlighting its ability to minimize false negatives. Recall is expressed as:

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

The F1-score, a harmonic mean of precision and recall, provides a balanced measure of a classifier's accuracy, particularly in scenarios where the class distribution is imbalanced. It is defined as:

$$F1\text{-score} = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (4)$$

These metrics allow a detailed assessment of the model's performance across various predictive aspects. CA offers an overall measure of correctness, precision ensures that false positives are minimized, and recall captures the model's ability to correctly identify true positives. The F1-score balances these two metrics, offering a single, interpretable value that encapsulates the trade-offs between precision and recall.

Additionally, a confusion matrix was employed to visualize and analyze the classifier's predictions for each class. This matrix provides a comprehensive breakdown of correct and incorrect predictions, enabling a deeper understanding of the model's strengths and weaknesses in distinguishing between the FFB classes.

4. Results and Discussion

This study aimed to classify oil palm FFBs into defective or non-defective categories using feature-based transfer learning pipelines combined with LR. Two experimental conditions were examined: pipelines relying on default LR hyperparameters and pipelines whose LR hyperparameters were systematically tuned. The goal was to identify whether tuning leads to performance gains and to evaluate how different CNN pipelines (EfficientNet-B0, EfficientNet-B4, ResNet152, and VGG16) compare in this application.

As summarized in Figure 4, classification accuracies were first evaluated using an untuned LR classifier. The numerical results of this scenario appear in Table 4. EfficientNet-B4 achieved the highest test accuracy (95.74%), followed by EfficientNet-B0, VGG16, and ResNet152. The comparatively lower accuracy observed for ResNet152 (88.30%) shows that deeper architectures may require more careful hyperparameter control. Precision and recall values in Table 4 highlight that EfficientNet-B4 attains consistently high F1-scores (96.15% for defective and 95.24% for non-defective), indicating a balanced performance. Other models, while still effective, showed pockets of misclassification. For instance, ResNet152 occasionally misclassified defective FFB as non-defective, a pattern possibly attributable to subtle FFB features or partial color overlaps.

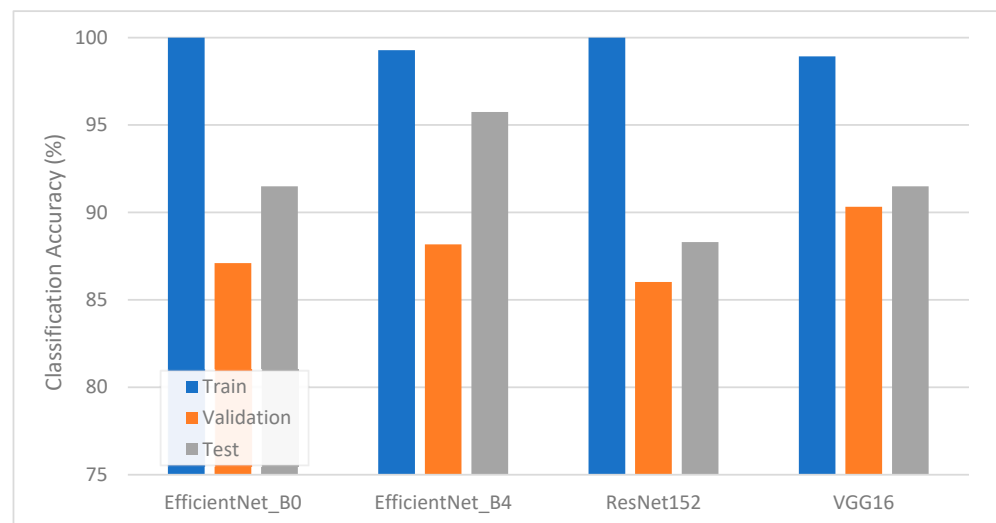


Figure 4. Classification accuracy of TL-LR pipelines without hyperparameter tuning.

Table 4. Performance measure of the test dataset obtained via TL-LR pipelines without hyperparameter tuning.

Pipeline	Class	Precision	Recall	F1-Score	CA
EfficientNet_B0-LR	0	94.12%	90.57%	92.31%	91.49%
	1	88.37%	92.68%	90.48%	
EfficientNet_B4-LR	0	98.04%	94.34%	96.15%	95.74%
	1	93.02%	97.56%	95.24%	
ResNet152-LR	0	88.89%	90.57%	89.72%	88.30%
	1	87.50%	85.37%	86.42%	
VGG16-LR	0	92.45%	92.45%	92.45%	91.49%
	1	90.24%	90.24%	90.24%	

Once the LR hyperparameters were tuned, further improvements were observed. Hyperparameter optimization revealed the best-performing LR configurations involved an $l2$ penalty for all pipelines, using the liblinear solver with C values of approximately 30.4 and 33.7 for EfficientNet-B0 and VGG16 features, respectively, and the saga solver with C values of approximately 13.5 and 0.07 for EfficientNet-B4 and ResNet152 features, respectively. Figure 5 depicts the training, validation, and test accuracies after tuning. The detailed precision, recall, and F1-scores for the test set appear in Table 5. EfficientNet-B4 again emerged as the top performer, with its test accuracy ascending to 96.81% and demonstrating F1-scores exceeding 96% for both classes. VGG16 also recorded a noteworthy gain, improving to a 94.68% test accuracy and around 95% F1-score for defective bunches. ResNet152 benefited substantially from tuning, posting a final accuracy of 93.62%. These results collectively confirm that matching an appropriate architecture with well-chosen LR hyperparameters can significantly reduce classification errors.

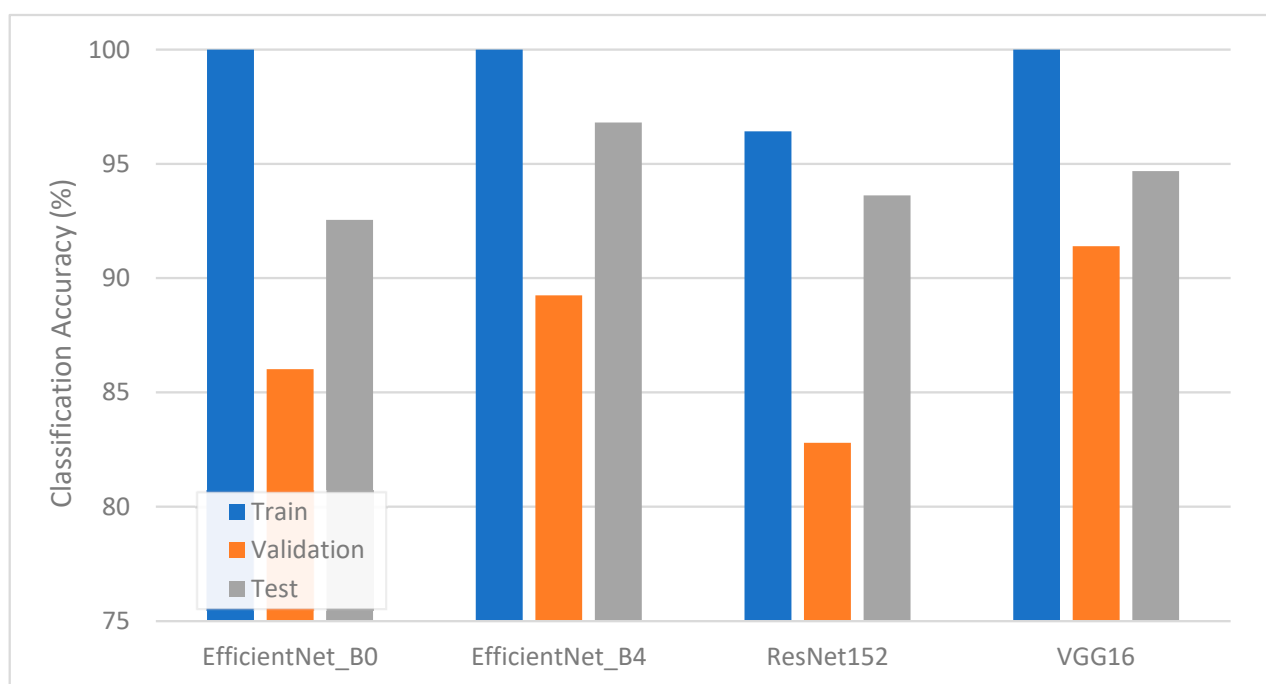


Figure 5. Classification accuracy of TL-LR pipelines with hyperparameter tuning.

Table 5. Performance measure of the test dataset obtained via TL-LR pipelines with hyperparameter tuning.

Pipeline	Class	Precision	Recall	F1-Score	CA
EfficientNet_B0-LR	0	96.00%	90.57%	93.20%	92.55%
	1	88.64%	95.12%	91.76%	
EfficientNet_B4-LR	0	98.08%	96.23%	97.14%	96.81%
	1	95.24%	97.56%	96.39%	
ResNet152-LR	0	96.08%	92.45%	94.23%	93.62%
	1	90.70%	95.12%	92.86%	
VGG16-LR	0	96.15%	94.34%	95.24%	94.68%
	1	92.86%	95.12%	93.98%	

The positive impact of hyperparameter tuning on LR is consistent with the research showing that penalty selection, solver choice, and regularization strength can alter the shape of the decision boundary in a feature space. Similar gains have been reported in image-based agricultural tasks [19], where subtle differences in color and texture require

carefully adjusted decision thresholds to ensure robust class separations. In this study, deeper networks, such as ResNet152, needed more precise tuning to overcome potential overfitting or underfitting, whereas an efficiently scaled architecture, like EfficientNet-B4, performed well, even under default LR settings, and then became more powerful when carefully optimized.

The relatively balanced performance of all four pipelines underscores that transfer learning can effectively extract discriminative features from FFB images. However, the small performance gap between the top (EfficientNet-B4) and other architectures hints that certain design elements, such as compound scaling, may be better suited to capturing differences in FFB appearance. Small color or texture shifts often signal the onset of defects, and a scaled approach appears to retain more relevant cues. Another factor influencing the results might be the differences in the input resolution: larger input sizes, as used by EfficientNet-B4, can preserve subtle low-level features essential for accurate differentiation.

Figure 6 illustrates the confusion matrices for EfficientNet-B4 with LR across the training, validation, and test sets. During training, the classifier achieves perfect separation between defective and non-defective classes. Although slight misclassifications appear in the validation with six defective images labeled as non-defective and four non-defective images mislabeled as defective, the model still correctly classifies over 88% of those samples. On the test set, 51 out of 53 defective FFBs are identified accurately, and 40 of the 41 non-defective samples are correctly recognized. These patterns suggest that although minor misclassifications remain, likely due to borderline FFB conditions, the combination of EfficientNet-B4's compound-scaling design and subsequent LR tuning yields a near-optimal solution, consistently identifying the key visual cues that differentiate defective from non-defective bunches.

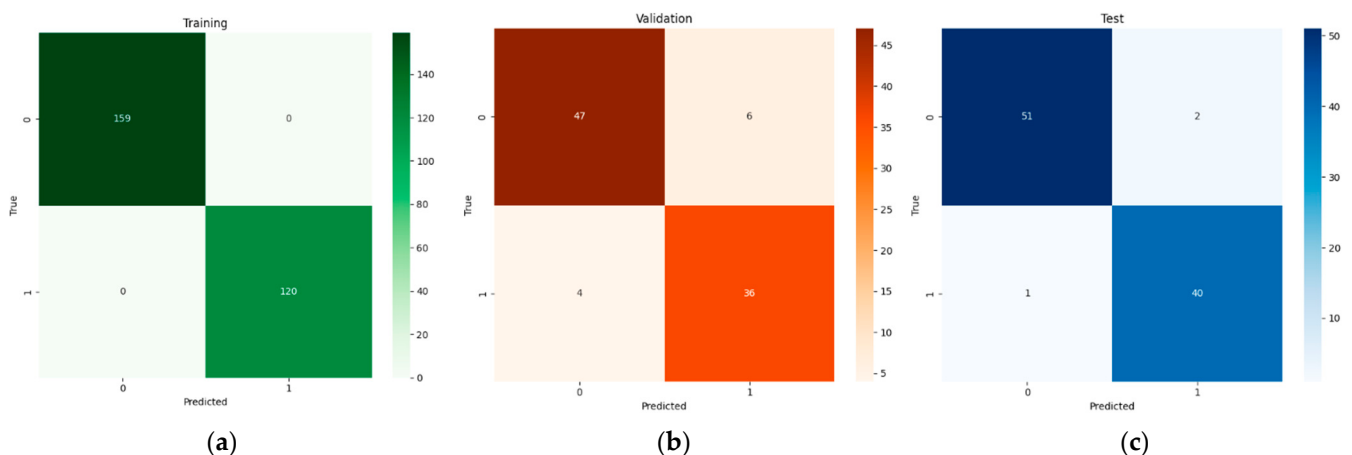


Figure 6. Confusion matrix of the efficientNet_B4-LR pipeline: (a) training and (b) validation (c) tests.

An examination of the misclassification patterns reveals key insights into model behavior across different CNN architectures. Based on our test results, ResNet152 demonstrates a higher false negative rate (15% of defective FFBs misclassified as non-defective) compared to EfficientNet-B4 (4% false negative rate), as shown in the confusion matrices. EfficientNet-B4's superior performance aligns with its compound scaling design, which systematically balances network width, depth, and resolution. Lighting conditions presented consistent challenges across all architectures, with test images showing a reduced classification accuracy under variable illumination conditions captured in the field dataset. The analysis revealed that FFBs with mixed visual characteristics posed classification difficulties, as evidenced by the misclassification patterns in our validation and test sets. Global average pooling inherently aggregates features across the entire image, which may reduce

the prominence of localized defect indicators. Future improvements could incorporate attention mechanisms to weight important image regions, multi-scale feature extraction to capture defects at different levels of detail, and targeted data augmentation to improve robustness under diverse field conditions.

To further evaluate the performance of our proposed methods, we compared them with several relevant studies, including the fine-tuned deep learning approaches of Kurniawan et al. [4], Mansour et al. [6], and Junior et al. [7], and the traditional feature engineering method of Rosbi et al. [9]. The results are summarized in Table 6. We can observe that the proposed feature-based transfer learning model, particularly the EfficientNet-B4 with tuned LR pipeline, achieves the highest classification accuracy (CA) among the compared methods, reaching 96.81%. This result surpasses the ~95% CA reported for fine-tuned networks by Kurniawan et al. [4] and the 93.68% CA from the hand-crafted feature approach by Rosbi et al. [9], while also comparing favorably to the mAP metrics reported for the object detection models [6,7], considering the potential differences in evaluation metrics. These results demonstrate the effectiveness of leveraging pre-trained models as feature extractors combined with tuned classifiers for accurate FFB defect classification.

Table 6. Performance comparison.

Authors	Methods	CA
Kurniawan et al. [4]	Fine-tuned AlexNet, ResNet, DenseNet	~95%
Rosbi et al. [9]	YOLOv4-Tiny Localization + Segmentation + Hand-crafted Features + MLP	93.68%
Proposed	EfficientNet-B4 (Feature extraction) + Tuned LR pipeline	96.81%

The computational efficiency of our feature-based transfer learning approach offers significant advantages for practical deployment in agricultural settings. Theoretical analysis reveals that our method requires a substantially lower computational overhead compared to conventional fine-tuning approaches. While fine-tuning necessitates gradient computation through entire networks with parameters ranging from 5.3M (EfficientNet-B0) to 138M (VGG16), our approach only trains the lightweight Logistic Regression classifier with feature dimensions of 512 to 2048, resulting in training speedups of 100–1000× and memory reductions from 50 to 80% [25]. The inference process is equally efficient, requiring only a single forward pass for feature extraction (0.39–15.5 GFLOPs depending on architecture) followed by minimal LR computation, enabling real-time classification on standard mobile processors. This computational profile makes our approach particularly suitable for resource-constrained agricultural environments, where offline processing capabilities, power efficiency, and hardware compatibility are critical considerations. The combination of high accuracy (96.81% with EfficientNet-B4) and low computational requirements positions our feature-based pipeline as a practical solution for large-scale FFB quality assessment systems, addressing the dual requirements of performance and deployability essential for industrial agricultural applications.

5. Conclusions

This research set out to develop and evaluate a pipeline for distinguishing defective and non-defective oil palm FFBs using feature-based transfer learning in combination with a LR classifier. The work addressed the need for reliable, automated grading methods in large-scale agricultural settings, where manual inspection can be time-consuming and prone to human error. By transferring the knowledge from pre-trained CNN models to an FFB classification scenario, the approach delivered high accuracy while requiring only moderate dataset sizes.

The methodology involved extracting convolutional features from four distinct CNN architectures, ranging from ultra-lightweight (EfficientNet-B0) and lightweight (EfficientNet-B4) to medium weight (ResNet152) and heavy weight (VGG16), and subsequently passing these features to a LR classifier. A 60:20:20 hold-out strategy was adopted for dataset splitting, with the final 20% of images reserved as a dedicated test set. To optimize the decision boundary in the feature space, the hyperparameter tuning of the LR classifier was employed, exploring various penalty functions, solver methods, and regularization strengths. Throughout these experiments, the Lightweight EfficientNet-B4 architecture emerged as the top performer. It achieved a classification accuracy above 96% and demonstrated consistent precision, recall, and F1-scores across both defective and non-defective classes.

These results imply that carefully scaled CNN architectures can capture the subtle textural and color variations that characterize different FFB maturity levels and potential defects. The effectiveness of the approach not only underscores the value of leveraging advanced deep learning architecture in agricultural contexts but also highlights the importance of systematically tuning downstream classifiers to enhance performance. The findings hold practical implications for palm oil producers, suggesting that an accurate and relatively resource-efficient solution could be deployed in real-world settings to streamline the inspection process.

Nevertheless, certain limitations should be noted. The hold-out method offers one perspective on model robustness; further cross-validation or more extensive data collection might yield finer-grained insights. In addition, borderline FFB conditions and partial defects remain challenging, and the future work might explore more elaborate data augmentation strategies or multi-view image acquisition to mitigate misclassification. Larger and more diverse datasets spanning different plantation locations or seasonal conditions could also be collected to bolster model generalization.

Notwithstanding these caveats, this study contributes to the existing research by verifying that transfer learning pipelines, when paired with an appropriately tuned LR classifier, can reliably classify FFBs in a practical, deployment-ready framework. Future extensions might apply this pipeline to other crop types or integrate complementary sensors (e.g., near-infrared imaging) to further refine the system's accuracy. Overall, the findings suggest a compelling avenue for advancing computer vision applications in agriculture, illustrating how state-of-the-art deep architectures and relatively simple classifiers can unite to deliver robust, high-accuracy solutions.

Author Contributions: Conceptualization, A.P.P.A.M. and Y.L.; methodology, Z.O.; formal analysis, Y.L. and Z.O.; investigation, Y.L. and S.A. resources, Y.C.; writing—original draft preparation, Y.L.; writing—review and editing, Y.L.; visualization, S.A.; supervision, A.P.P.A.M.; project administration, Y.C. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Research Development Fund (No. RDF-21-01-028) and Project for Centre of Excellence for Syntegrative Education (COESE2324-01-07) of Xi'an Jiaotong-Liverpool University.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The dataset is publicly available via the following link: <https://zenodo.org/records/11114885> (accessed on 16 February 2025).

Conflicts of Interest: The authors declare no conflicts of interest.

References

- Voora, V.; Bermudez, S.; Farrell, J.; Larrea, C.; Luna, E. *Palm Oil Prices and Sustainability*; Sustainable Commodities Marketplace Series; IISD: Winnipeg, MB, Canada, 2023.
- Gregory, M. *Palm Oil Production, Consumption and Trade Patterns: The Outlook from an EU Perspective*; Fern: Brussels, Belgium, 2022.
- Ibrahim, Z.; Sabri, N.; Isa, D. Palm Oil Fresh Fruit Bunch Ripeness Grading Recognition Using Convolutional Neural Network. *J. Telecommun. Electron. Comput. Eng. (JTEC)* **2018**, *10*, 109–113.
- Kurniawan, R.; Samsuryadi, S.; Mohamad, F.S.; Wijaya, H.O.L.; Santoso, B. Classification of palm oil fruit ripeness based on AlexNet deep Convolutional Neural Network. *SINERGI* **2025**, *29*, 207–220. [[CrossRef](#)]
- Kurniawan, R.; Samsuryadi, S.; Mohamad, F.S.; Wijaya, H.O.L.; Santoso, B. Advancing palm oil fruit ripeness classification using transfer learning in deep neural networks. *Bull. Electr. Eng. Inform.* **2025**, *14*, 1126–1137. [[CrossRef](#)]
- Mansour, M.Y.M.A.; Dambul, K.D.; Choo, K.Y. Object Detection Algorithms for Ripeness Classification of Oil Palm Fresh Fruit Bunch. *Int. J. Technol.* **2022**, *13*, 1326–1335. [[CrossRef](#)]
- Adeta, F., Jr.; Suharjito. Video based oil palm ripeness detection model using deep learning. *Heliyon* **2023**, *9*, e13036. [[CrossRef](#)]
- Suharjito; Junior, F.A.; Koeswandy, Y.P.; Debi; Nurhayati, P.W.; Asrol, M.; Marimin. Annotated Datasets of Oil Palm Fruit Bunch Piles for Ripeness Grading Using Deep Learning. *Sci. Data* **2023**, *10*, 72. [[CrossRef](#)] [[PubMed](#)]
- Rosbi, M.; Omar, Z.; Khairuddin, U.; Majeed, A.P.; Bakar, S.A. Machine learning for automated oil palm fruit grading: The role of fuzzy C-means segmentation and textural features. *Smart Agric. Technol.* **2024**, *9*, 100691. [[CrossRef](#)]
- Suharjito; Elwirehardja, G.N.; Prayoga, J.S. Oil palm fresh fruit bunch ripeness classification on mobile devices using deep learning approaches. *Comput. Electron. Agric.* **2021**, *188*, 106359. [[CrossRef](#)]
- Mamat, N.; Othman, M.F.; Abdulghafor, R.; Alwan, A.A.; Gulzar, Y. Enhancing Image Annotation Technique of Fruit Classification Using a Deep Learning Approach. *Sustainability* **2023**, *15*, 901. [[CrossRef](#)]
- Omar, Z.; Abdul Majeed, A.P.; Rosbi, M.; Ghazalli, S.A.; Selamat, H. Outdoor oil palm fruit ripeness dataset. *Data Brief* **2024**, *55*, 110667. [[CrossRef](#)] [[PubMed](#)]
- Zhuang, F.; Qi, Z.; Duan, K.; Xi, D.; Zhu, Y.; Zhu, H.; Xiong, H.; He, Q. A comprehensive survey on transfer learning. *Proc. IEEE* **2020**, *109*, 43–76. [[CrossRef](#)]
- Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. *Commun. ACM* **2017**, *60*, 84–90. [[CrossRef](#)]
- Zhang, W.; Li, R.; Zeng, T.; Sun, Q.; Kumar, S.; Ye, J.; Ji, S. Deep Model Based Transfer and Multi-Task Learning for Biological Image Analysis. In Proceedings of the KDD '15: 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Sydney, Australia, 10–13 August 2015; pp. 1475–1484. [[CrossRef](#)]
- Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely connected convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4700–4708.
- Luo, Y.; Liu, X.; Song, R.; Abdul Majeed, A.P.; Zhang, F.; Xu, Y.; Tan, A.; Chen, W. Enhancing Bone Fracture Detection: A Feature-Based Transfer Learning Approach Using DenseNet with SVM. In Proceedings of the 2nd International Conference on Intelligent Manufacturing and Robotics, ICIMR 2024, Suzhou, China, 22–23 August 2025; pp. 362–367.
- Kumar, J.L.M.; Rashid, M.; Musa, R.M.; Razman, M.A.M.; Sulaiman, N.; Jailani, R.; Majeed, A.P.A. The classification of EEG-based wink signals: A CWT-transfer learning pipeline. *ICT Express* **2021**, *7*, 421–425. [[CrossRef](#)]
- Luo, Y.; Chen, Y.; Abdul Majeed, A.P. Optimizing poultry disease classification: A feature-based transfer learning approach. *Smart Agric. Technol.* **2025**, *10*, 100856. [[CrossRef](#)]
- Luo, Y.; Lu, X.; Chen, Y.; Andresen, J.; Maroto-Valer, M. Liquid Natural Gas Cold Energy Recovery for Integration of Sustainable District Cooling Systems: A Thermal Performance Analysis. *Inventions* **2023**, *8*, 121. [[CrossRef](#)]
- Luo, Y.; Jagtap, S.; Trollman, H.; Garcia-Garcia, G.; Liu, X.; Abdul Majeed, A.P. Optimizing Industrial Etching Processes for PCB Manufacturing: Real-Time Temperature Control Using VGG-Based Transfer Learning. In Proceedings of the 2nd International Conference on Intelligent Manufacturing and Robotics, ICIMR 2024, Suzhou, China, 22–23 August 2025; pp. 353–361.
- Tan, M.; Le, Q. Efficientnet: Rethinking model scaling for convolutional neural networks. In Proceedings of the International Conference on Machine Learning, Long Beach, CA, USA, 9–15 June 2019; pp. 6105–6114.
- He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.

24. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
25. Razavian, A.S.; Azizpour, H.; Sullivan, J.; Carlsson, S. CNN Features Off-the-Shelf: An Astounding Baseline for Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW 2014), Columbus, OH, USA, 23–28 June 2014; pp. 806–813.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.