STATISTICS, OPTIMIZATION AND INFORMATION COMPUTING
Stat., Optim. Inf. Comput., Vol. **00**, March 0000, pp 1–25.
Published online in International Academic Press (www.IAPress.org)

STEM-Based Analysis of Avocados (*Persea americana Mill.*) Leaf Classification using Deep Convolutional Neural Networks

Rina Sugiarti Dwi Gita [1,3], Zainur Rasyid Ridlo [4*], Rifki Ilham Baihaki [5], Firma Nur Muttakin [4], Dafik[2,5], Arika Indah Kristiana[3]

[1]*Department of Learning Technology, Universitas PGRI Argopuro Jember, Jember, Indonesia*
[2]*Department of Postgraduate Mathematics Education, University of Jember, Jember, Indonesia*
[3]*Department of Mathematics Education, University of Jember, Jember, Indonesia*
[4]*Department of Science Education, University of Jember, Jember, Indonesia*
[5]*PUI-PT Combinatorics and Graph, CGANT-University of Jember, Jember, Indonesia*

Abstract Accurate identification of avocado (*Persea americana* Mill.) varieties based on leaf morphology is essential for precision agriculture, breeding programs, and field-based decision support, yet remains challenging due to subtle inter-varietal similarities and environmental variability. This study proposes a STEM-based deep learning framework for avocado leaf classification using advanced Deep Convolutional Neural Networks (DCNNs). A dataset comprising 7,000 leaf images from seven avocado varieties was collected under diverse field conditions and systematically annotated with environmental and morphological metadata. Three architectures namely EfficientNetV2, ConvNeXt, and ViT-Hybrid CNN were evaluated using 10-fold cross-validation to ensure robust performance estimation. Statistical rigor was reinforced through paired t-tests and 95% confidence intervals for accuracy, precision, recall, and F1-score. Experimental results demonstrate that EfficientNetV2 achieved the highest classification accuracy of 98.71% (95% CI: [98.45%, 98.97%]), significantly outperforming competing architectures ($p < 0.05$), while maintaining favorable computational efficiency. Explainable AI (XAI) analysis using Grad-CAM revealed that the proposed models consistently focused on biologically relevant leaf regions, such as vein structures and leaf margins, enhancing model interpretability and practical relevance. Further evaluation under unconstrained real-world conditions showed only a modest performance degradation, confirming model robustness. Comparative analysis with lightweight architectures (MobileNetV3, ShuffleNet, and EfficientNetV2-Lite) highlighted clear trade-offs between classification accuracy and deployment feasibility for edge-based applications. Overall, the proposed framework demonstrates that RGB-based DCNN models can achieve accurate, interpretable, and deployable avocado variety classification using low-cost imaging devices, offering a scalable solution for real-world agricultural applications.

Keywords ConvNeXt, DCNN, EfficientNetV2, Persea americana Mill, and Vision Transformer (ViT)-Hybrid CNN.

AMS 2010 subject classifications 68T07

DOI: 10.19139/soic-2310-5070-3124

1. Introduction

Food security is a fundamental aspect of national development because it is directly related to the quality of life, public health, and economic stability of a country [1, 2]. In the current global context, the issue of food security has become increasingly urgent due to three main factors: climate change, rapid population growth, and limited agricultural land. Significant and sustained climate change can affect rainfall patterns, increase the frequency of natural disasters, and degrade ecosystems making agricultural productivity unpredictable [3]. Avocado (*Persea americana Mill.*) is one of the horticultural crops that has great potential to maintain national food security, but its utilization has not been done optimally. The advantages of avocado that make it ideal to support national food security are rich in nutrition and have strong economic value [4, 5]. Avocado is included in superfoods that contain

*Correspondence to: Zainur Rasyid Ridlo (Email: zainur@fkip.unej.ac.id). Department of Science Education, University of Jember. 37 Kalimantan Road, Jember, East Java Province, Indonesia (68131).

ISSN 2310-5070 (online) ISSN 2311-004X (print)

a lot of potassium, vitamins E and K, and unsaturated fats that can improve heart health. Therefore, avocado is one of the strategic commodities. This commodity can improve welfare, support food diversification and increase the resilience of the national food system in the long term.

(a) (b) (c)

Figure 1. Illustration of an avocado tree with its fruits and leaves

Figure 1 shows (a) a representation of an avocado tree, (b) a visualization of an avocado fruit, and (c) a depiction of an avocado leaf. Leaves play a crucial role in producing quality avocado fruit because they are directly related to the efficacy of photosynthesis. Morphologically, leaves have shape, size, texture, and in particular green pigmentation with chlorophyll concentration within them. This leaf morphology not only helps differentiate between varieties but also serves as an indicator of the plant's physiological capacity [6, 7]. In the past, avocado variety classification was carried out by horticulturists through visual assessment. However, this method was generally subjective and prone to inconsistencies. Recent technological advances have enabled more objective and quantitative assessment of leaf characteristics through digital image analysis and machine learning techniques [8, 9]. In this context, machine learning provides a data-driven framework for the automatic and objective classification of avocado leaves, leveraging image processing and advanced classification algorithms to achieve high precision and reliability [10, 11].

The application of machine learning, namely Deep Convolutional Neural Networks (DCNN), not only prevents human errors but also improves operational efficiency, facilitates rapid classification of leaf specimens and enhances the precision of variety identification, which is crucial for informed scientific decision-making in the field of avocado breeding and cultivation [15][16]. This study uses the STEM (Science, Technology, Engineering, and Mathematics) framework to ensure a comprehensive and structured analysis in exploring the morphological properties of avocado leaves[12][13]. This study is grounded in a STEM-based computational thinking framework, consistent with previous findings highlighting the effectiveness of deep learning approaches in supporting analytical reasoning and scientific problem-solving [56]. The scientific aspect deals with the morphology and characterization of avocado leaves. The technological and engineering aspects will include the application of advanced computing devices and image processing methods for analysis and explainable artificial intelligence (XAI) techniques are integrated to visualise specific leaf contours. In addition, this study uses three DCNN architectures, namely EfficientNetV2, ConvNeXt, and Vision Transformer (ViT)-Hybrid CNN and three Lightweight models (MobileNetV3, ShuffleNet, EfficientNetV2-Lite) to classify leaves accurately and consistently. On the other hand, The mathematical aspect plays a critical role in developing robust classification models by integrating statistical evaluation, feature extraction techniques, and DCNN-based optimization strategies to achieve precise and reliable leaf attribute classification [17, 18, 26].

2. Research Methods

To ensure dataset standardization while reflecting real-world field conditions, avocado leaf images were acquired under diverse environmental settings, including variations in ambient temperature, relative humidity, and illumination intensity commonly observed in orchard environments. During image acquisition, temperature and relative humidity were measured using a portable digital agro-meteorology logger, while illumination levels were recorded using a handheld lux meter. All images were captured under natural lighting conditions using a smartphone camera (108 MP, focal length 1.8 mm). To enhance data transparency and reproducibility, relevant acquisition metadata were systematically recorded, including camera specifications, illumination intensity, geographic coordinates, leaf growth stage, and visually observed stress conditions. Each sampling location was georeferenced to preserve environmental context, and leaf developmental stage and stress status were annotated based on in-field observations. Incorporating controlled environmental variability and detailed

metadata documentation reduces bias associated with homogeneous acquisition conditions and supports improved generalization of the proposed model to unseen field environments.

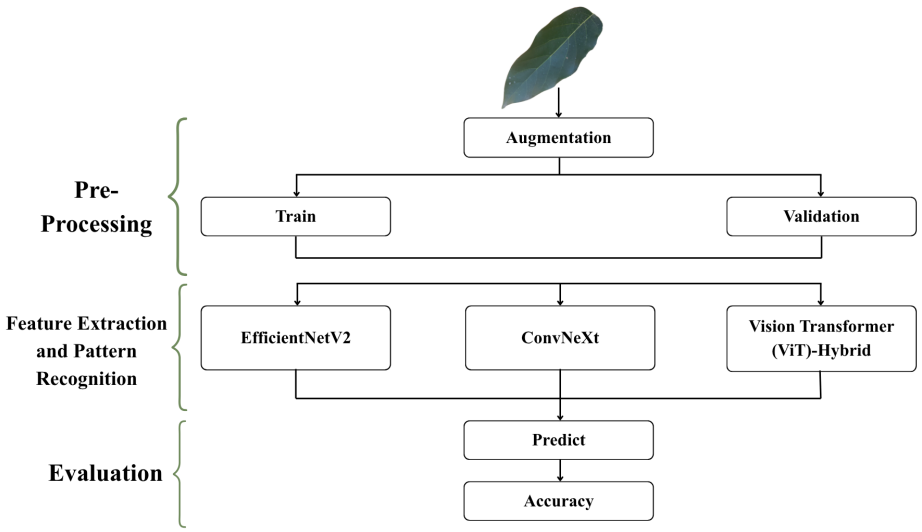

Figure 2. The steps of avocado leaf classification using DCNN

Figure 2 shows the proposed framework method and consists of three sequential stages. The first stage, (i) **Preprocessing** is the process of preparing raw input data by applying operations such as cleaning, normalization, resizing, and enhancement to ensure consistency and reduce noise. The dataset is randomly divided into three parts with a percentage of 70% for training, 15% for validation, and 15% for testing to ensure the model is balanced during the learning and evaluation process [18]. In addition, data augmentation techniques are also carried out to improve model generalization and reduce overfitting. The data augmentation carried out includes random rotation ($\pm 20°$), horizontal and vertical flips, scaling and shifting, and random brightness and contrast adjustments. During this process, all input image sizes are resized to match the model's input dimensions of 224×224 pixels. This aims to ensure uniformity and compatibility with the DCNN architecture. Next, pixel values will be normalized to the range [0, 1] for numerical stability and to maintain the ImageNet normalization scheme or to the range [1, 1][19]. Additionally, the class labels corresponding to the seven avocado leaf varieties were converted using one-hot encoding to produce an output vector that was mapped to a softmax activation layer.

The second stage of this process is **(ii) Extraction of features and recognition of patterns**. During this phase, the convolutional layers of a deep convolutional neural network (DCNN) architecture function as hierarchical feature extractors that gradually transform raw pixel information into abstract and meaningful feature maps. Low-level visual cues, such as edges, color gradients, and texture orientations, are captured by early convolutional layers. Next, deeper layers serve to encode high-level structural attributes and spatial correlations to distinguish between different types of avocado leaves. At this stage, a padding matrix is introduced to preserve spatial dimensionality and prevent the loss of edge information during the convolution operation. The extracted features are further processed by fully connected layers and softmax activation layers, which facilitate pattern recognition and decision mapping. This enables the network to associate specific feature combinations with corresponding class labels [54, 55].

Finally, the third stage as **(iii) Evaluation**. In this stage, the trained DCNN model is evaluated using a previously unseen test dataset to measure its ability to generalise beyond the training sample. A range of evaluation metrics are employed to provide a comprehensive assessment of the classification performance, including accuracy, precision, recall, F1 score and the confusion matrix. This enables the identification of potential errors in classification process. As illustrated in Figure 2.

2.1. Digital Image Processing

An image can be mathematically represented as a two-dimensional function $f(x, y)$, where x and y denote the spatial coordinates in a plane, and the value of f at any coordinate pair (x, y) corresponds to the amplitude, which is commonly referred to as the intensity or grayscale level at that point, see Figure 3. When the spatial coordinates (x, y) as well as the intensity values f are finite and quantized into discrete levels, the image is no longer continuous but instead becomes a collection of discrete samples. In this case, the image is categorized as a digital image, since both its spatial domain and intensity range are represented in discrete form [25].

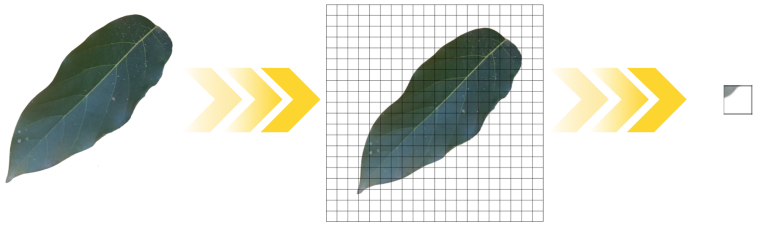

Figure 3. Image extraction for DCNN process

Image feature extraction in Deep Convolutional Neural Networks (DCNN) refers to the transformation of raw pixel data into a hierarchical representation of informative features that capture structural patterns within an image see Figure 3.

2.2. Artificial Neural Networks Architecture

The field of data mining frequently employs Artificial Neural Networks (ANNs) which consist of interconnected nodes analogous to neurons in the human brain. Each node, or perceptron, acts as a processing unit that integrates signals transmitted through weighted connections. These weights determine the strength of interactions between neurons and regulate the flow of encoded information [27, 28, 29]. An ANN typically comprises three layers: an input layer that receives external data, one or more hidden layers where computations and pattern learning occur, and an output layer that produces predictions or classifications, as illustrated in Figure 4. Mathematically, neuron connections are represented by a weight matrix $W = (w_{ij})$. For a neuron Y_j, the input is the dot product of the input vector $x = (x_1, x_2, \ldots, x_n)$ and the weight vector w_j: $\hat{Y} = \sum_{i=1}^{n} x_i w_{ij}$. A bias term (β) can be added by augmenting the input vector with $x_0 = 1$, yielding: $\hat{Y} = \beta_j + \sum_{i=1}^{n} x_i w_{ij}$.

The basic operation of neural networks is to sum its weight times input signal and apply the activation function. There are some types of activation functions are: (*i*) Linear activation function $\hat{Y} = f(x) = ax + b$, when $a = 1, b = 0$, it is an identity; (*ii*) Binary step activation function with threshold θ; $\hat{Y} = f(x) = \begin{cases} 1 & \text{if } x \geq \theta \\ 0 & \text{if } x < \theta \end{cases}$; (*iii*) Binary sigmoid activation function $\hat{Y} = f(x) = \frac{1}{1+e^{-\sigma x}}$, and $f'(x) = \sigma f(x)[1 - f(x)]$; (*iv*) Bipolar sigmoid activation function $\hat{Y} = g(x) = 2f(x) - 1 = \frac{1-e^{-\sigma x}}{1+e^{-\sigma x}}$, and $g'(x) = \frac{\sigma}{2}[1 + g(x)][1 - g(x)]$; (*v*) Hyperbolic tangent activation function (Tanh) $\hat{Y} = h(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}$, and $h'(x) = [1 + h(x)][1 - h(x)]$

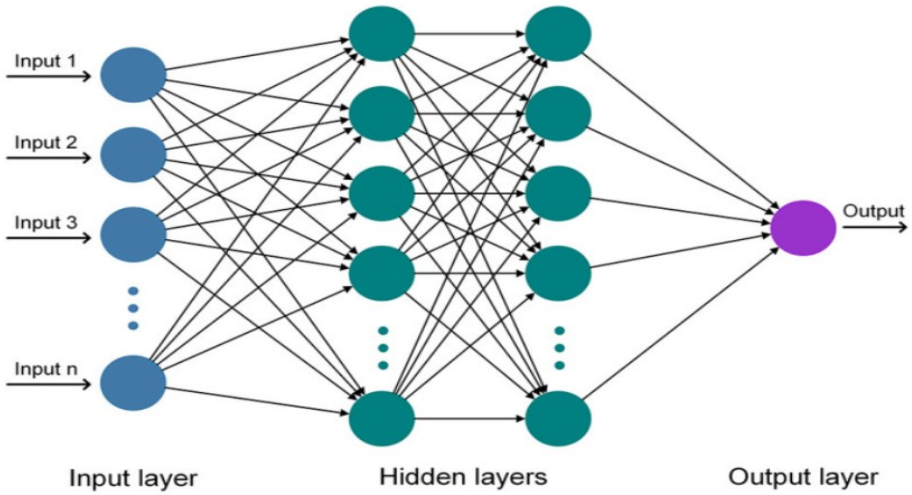

Figure 4. The fully connected Artificial Neural Networks architecture

2.3. DCNN Algorithm

For developing a good model of ANN, we use the following multilayer Perceptron algorithm. The modification of ANN is proposed to development Deep Convolutional Neural Networks (DCNNs) architechtures. Deep Convolutional Neural Networks (DCNNs) are a class of neural networks specifically designed to process data with a grid-like structure, such as images. A CNN is characterized by the presence of layers that perform convolution operations to automatically extract features from the input. The process begins by partitioning the input image into multiple small, often overlapping, regions, as illustrated in Figure 5. This localized division allows the network to focus on smaller details within the image while preserving spatial relationships [30, 31].

General DCNN Algorithm

Step 0. Initialize parameters: Weights (w), bias (β), learning rate (α), and error goal.

Step 1. Feedforward stage $h_i = \beta + \sum_{i=0}^{n} x_i \cdot w_{i,j}$, where n is the number of inputs.

Step 2. Calculate the output $y_{in_j} = \beta + \sum_{i=0}^{n} w_{i,j} \cdot h_i$

Step 3. Output activation using log-sigmoid $\hat{y}_m = \frac{1}{1+e^{-y_{in_m}}}$

Step 4. Calculate the output error $\delta k_i = (t - \hat{y}_i)^2$, and $MSE = \frac{\sum_{i=0}^{n}(t-\hat{y}_i)^2}{n}$

Step 5. Calculate the backpropagation error between output layer and hidden layer $\delta j'_j = \beta \sum_{i=0}^{n} w_{i,j} \cdot \delta k_i$

Step 6. Calculate the backpropagation error between hidden layer and input layer $\delta j'_j = \sum_{i=0}^{n} w_{i,j} \cdot \delta k_i$

Step 7. Update weight and bias $w_{\text{new}} = w_{\text{old}} + \alpha \cdot \delta k_i \cdot h_i, \beta_{\text{new}} = \beta_{\text{old}} + \alpha \cdot \delta j_j \cdot x_i$

Step 8. Checking the termination criteria (epoch reaches the maximum number or $MSE \leq$ error goal)

Step 9. If not, go back to Step 1.

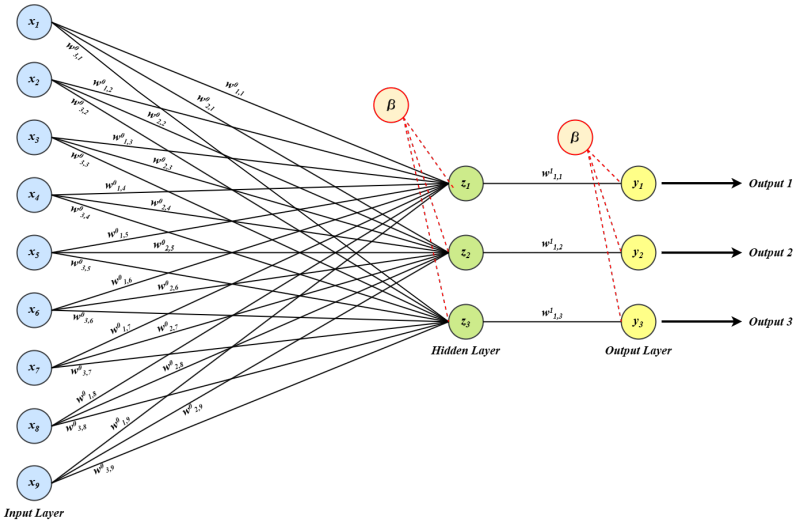

Figure 5. Architecture of the CNN

In this study, three deep convolutional neural network (DCNN) architectures were applied including **EfficientNetV2**, **ConvNeXt**, **Vision Transformer (ViT)-Hybrid CNN** and three Lightweight models (**MobileNetV3**, **ShuffleNet**, **EfficientNetV2-Lite**). Paired t-tests were performed using accuracy values obtained from the 10-fold cross-validation to evaluate whether performance differences between models were statistically significant. A significance level of 0.05 was adopted. In addition to statistical significance testing, 95% confidence intervals (CI) were computed for accuracy, precision, and recall based on the results of the 10-fold cross-validation to quantify performance

EfficientNetV2 is an advanced DCNN architecture designed to optimise the balance between accuracy, computational efficiency and scalability. By integrating compound scaling and introducing *Fused-MBConv* blocks, EfficientNetV2 accelerates the training process while maintaining a high level of representational capacity. In addition, EfficientNetV2 has three main differences compared to its predecessor, namely the use of a smaller expansion ratio to reduce memory load, prioritization of small kernels (3×3) with additional layers to maintain the receptive field, and the elimination of the last step with stride $\bar{1}$ to reduce the number of parameters and memory access complexity [21]. Studies have demonstrated its superior performance in medical and industrial image processing tasks, achieving high accuracy and robustness at a lower computational cost, particularly when combined with transfer learning techniques for small-scale datasets. The illustration of EfficientNetV2 architecture show on Figure 6.

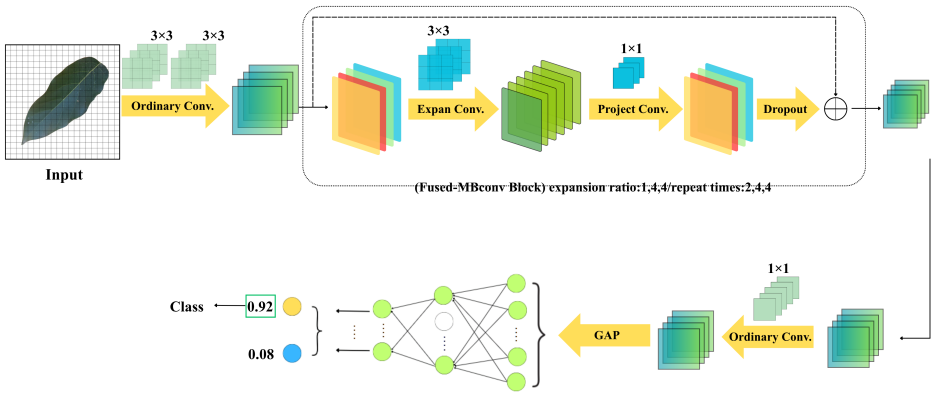

Figure 6. EfficientNetV2 architecture[16]

The second architecture is ConvNeXt, is a modern convolutional framework that reinterprets classical CNN design principles through the lens of Vision Transformers. By incorporating *layer normalisation*, *depthwise convolution*, and *residual connections*, ConvNeXt bridges the gap between convolutional and transformer-based models [19, 22, 23]. The illustration of Architecture of the ConvNeXt architecture show on Figure 7.

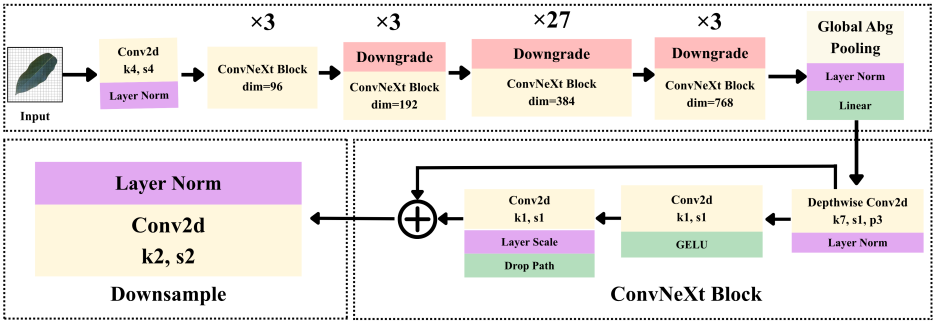

Figure 7. Architecture of the ConvNeXt[19]

The third architecture is Vision Transformer (ViT)-Hybrid CNN, combines the local feature extraction power of CNNs with the global attention mechanism of Transformers. This hybrid framework employs *progressive tokenisation* and *convolutional embeddings* to overcome ViT's reliance on large-scale pre-training[24, 57]. The illustration of Architecture of the ConvNeXt architecture show on Figure 8.

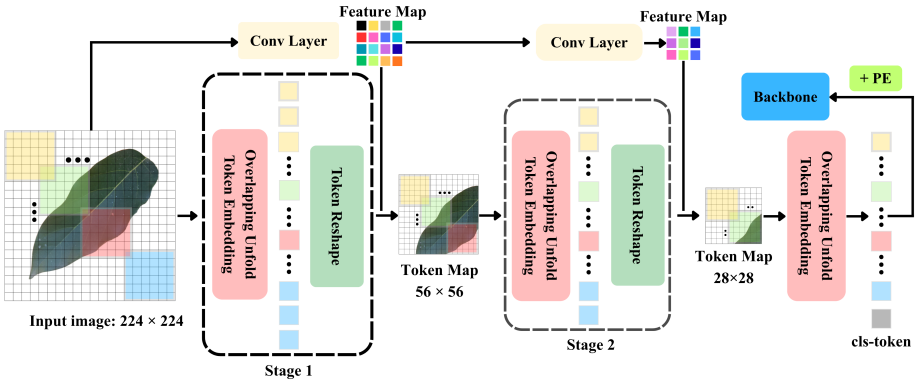

Figure 8. Architecture of the Vision Transformer (ViT)-Hybrid CNN

To ensure methodological transparency and reproducibility, full implementation details of the proposed system are provided in this section, including model configurations, hyperparameters, preprocessing procedures, augmentation settings, and computational environment. Table 1 presents the training hyperparameter configuration of the proposed model.The training process utilized the Adam optimization algorithm, categorical cross-entropy loss, dropout regularization, and He-normal weight initialization. Image inputs were standardized through uniform resizing prior to training. All models were implemented using TensorFlow/Keras and trained under identical conditions to enable fair comparison across architectures[60, 61]. The EfficientNetV2 backbone was configured using its default compound scaling strategy, with Fused-MBConv blocks applied in the early stages and regular MBConv blocks in later stages [58, 59]. The input resolution was set to 224×224 pixels for all models. For comparison, ConvNeXt and the ViT-Hybrid CNN were implemented with their standard base configurations to avoid architectural bias. Kernel size was fixed at 3×3 for all convolutional stages in EfficientNetV2.

Table 1. Training Hyperparameters in This Study

Parameter	Value
Learning rate	0.0001
Optimizer	Adam
Batch size	32
Epochs	100
Loss function	Categorical cross-entropy
Dropout	0.3
Weight initialization	He-normal
Input resolution	224×224

Table 2 presents the data preprocessing and augmentation steps applied in this study. Image normalization was performed using a standard reference distribution to align input data with the pretrained model requirements. All images were resized to a uniform spatial resolution prior to training. To enhance data diversity and improve model generalization, several augmentation techniques were applied, including horizontal flipping, rotation, brightness and contrast adjustment, as well as random zooming. These operations were designed to simulate variations commonly encountered in real-world image acquisition conditions[62, 63].

Table 2. Data Augmentation and Preprocessing Steps

Step	Parameter
Normalization	ImageNet mean/std
Resizing	224×224 px
Horizontal flip	$p = 0.5$
Rotation	$\pm 20^{\circ}$
Brightness adjustment	0.8–1.2
Contrast adjustment	0.8–1.2
Random zoom	0.95–1.05

The hardware and software configuration summarized in Table 3 was selected to ensure a reliable and reproducible experimental environment. The use of a standard computing platform with integrated processing capabilities reflects a realistic and accessible setup for model development and evaluation. Employing a widely adopted operating system and a high-level programming language facilitates compatibility and ease of replication across different research settings.

Table 3. Hardware and Software Configuration

Component	Specification
GPU	Intel UHD Graphic
CPU	Intel Core I5-1035G1
RAM	8 GB
OS	Windows 11
Python	3.10
TensorFlow	2.11
CUDA/cuDNN	matching version

2.4. Environmental Conditions and Datasets

The model demonstrated stable performance across different microclimate conditions, indicating that the inclusion of environmental variation during dataset collection contributed to improved robustness. This finding suggests that standardized acquisition procedures and environmental metadata are essential for real-world deployment of computer-vision-based variety identification [64]. These acquisition details, including camera specification, field illumination, growth stage, and stress status, contribute to model robustness by ensuring that the dataset reflects realistic field variations rather than only controlled imaging conditions[65]. Accordingly, the

field data acquisition parameters summarized in Table 4 provide critical contextual support for these findings by documenting the imaging conditions and environmental factors under which the dataset was collected. Additionally, the representative samples summarized in Table 5 were selected to balance intra-class variability and inter-class consistency. Controlled image acquisition conditions and careful sample selection ensure that the dataset reflects meaningful morphological differences among avocado varieties while minimizing bias arising from illumination, scale, or viewpoint inconsistencies.

Table 4. Field Data Acquisition Parameters

Category	**Variable**	**Value / Range**	**Instrument**	**Notes**
Camera	Sensor model	Sony IMX682	Smartphone camera	1/1.7" CMOS sensor
	Resolution	3000×4000 px	Native setting	No digital zoom applied
	File format	JPEG	Default setting	Stored without recompression
Environmental Sensors	Air temperature	26–33°C	Digital thermometer	Ambient temperature during capture
	Air humidity	65–88%	Digital hygrometer	Relative humidity measurement
	Light intensity	350–900 lux	LX-1108 lux meter	Measured at leaf surface
Location	Coordinates	-8.159°, 113.702°	Smartphone GPS	Jember, East Java
	Orchard type	Smallholder farm	Field observation	Rainfed plantation
Leaf Stage	Growth stage	Young / mature	Visual assessment	Expert phenological annotation
Stress Status	Disease symptoms	Mild–severe	Field inspection	Lesion and chlorosis observed
	Environmental stress	Drought / normal	Soil observation	Dry-season sampling

Table 5. Overview of Representative Avocado Leaf Samples

Avocado Variety	**Notes**
Hass	Representative samples captured under consistent illumination conditions, including both young and mature leaves with stressed and non-stressed appearances.
Clara	Samples illustrate characteristic vein variations observed across different growth stages and stress conditions.
Aligator	Leaves selected to reflect a similar size range and morphology under uniform illumination.
Marcus	Examples include field-level variations resulting from natural environmental conditions.
Sujuavo	Representative leaves captured with comparable lighting and consistent camera viewpoints.
Yamagata	Samples obtained using similar camera angles to ensure visual consistency across classes.
Pluwang	Balanced visual examples selected to minimize bias and represent typical morphological traits.

To provide a transparent overview of the dataset composition, we also included representative sample images from each class with similar distribution in terms of viewpoint, illumination, and growth stage. These visual samples allow readers to observe inter-class morphological differences as well as intra-class variations, thereby enhancing the interpretability of the dataset characteristics [66]. By presenting comparable visual examples across

all classes, the dataset section enables better understanding of the inherent complexity of the leaf samples and supports reproducibility for future comparative studies.

2.5. Expert-Based Morphological Validation of XAI Outputs

To assess the biological plausibility of the XAI results, a qualitative expert-based validation was conducted[67]. Selected Grad-CAM visualizations were reviewed by horticulture experts specializing in avocado morphology. The experts examined whether the highlighted regions corresponded to known varietal characteristics such as leaf venation patterns, leaf margins, and overall leaf shape. This validation aimed to ensure that the model's decision-making process aligns with established botanical knowledge[68].

2.6. Deployment Prototype on Smartphones

To demonstrate real-world deployment feasibility, a prototype Android application was developed to perform avocado leaf classification using the trained deep learning model [69, 70]. Figure 9 shows that Android application development has several stages starting from data collection, selecting a CNN model, creating coding and the application development process. The application allows users to capture or upload leaf images and receive real-time classification results. This prototype serves as a proof-of-concept for edge-based deployment under practical conditions. To assess real-world applicability, additional experiments were conducted using images captured in unconstrained conditions, including varying illumination, background clutter, and partial leaf occlusion[71]. A subset of field-relevant images was evaluated using the developed Android application to measure indicative accuracy and inference time. To evaluate the practical utility of the proposed model, additional testing was conducted using real-world images captured under unconstrained field conditions[72]. The performance of the model was compared between the controlled dataset used for training and a subset of real-world images acquired via a smartphone-based Android application[73].

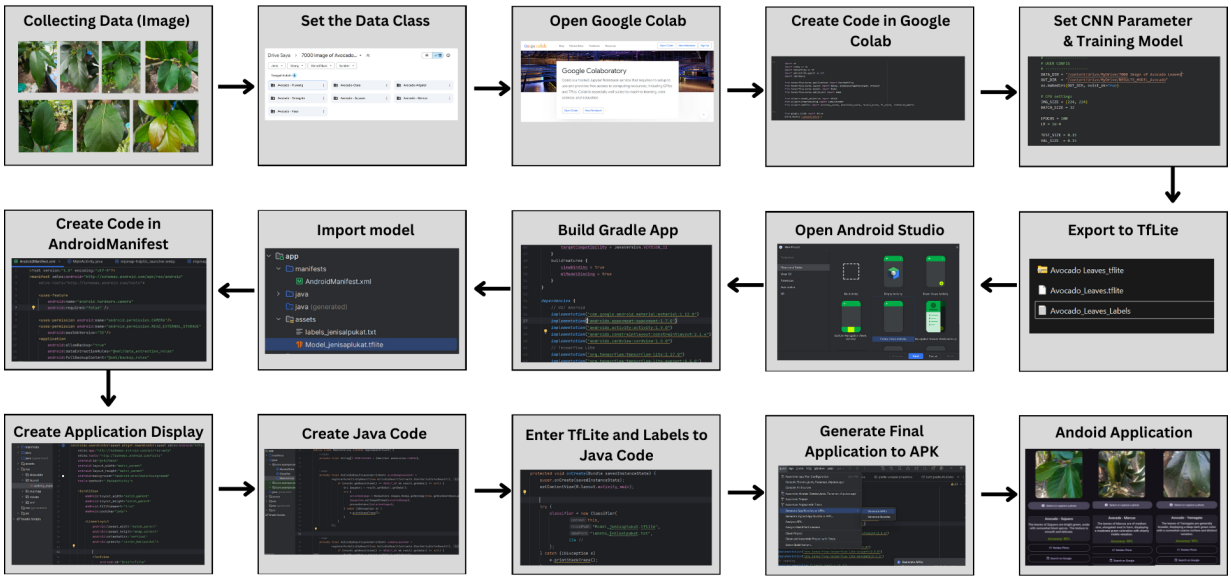

Figure 9. Android Application Design Process for Avocado Leaf Classification

3. Result and Discussion

This study evaluates three state-of-the-art deep convolutional neural network (DCNN) architectures representing recent advances in computer vision. EfficientNetV2 employs compound scaling to balance network depth, width, and input resolution, enabling high accuracy with computational efficiency. ConvNeXt modernizes convolutional design through simplified blocks, improved normalization, and large-kernel convolutions to achieve performance comparable to Vision Transformers while retaining CNN efficiency. In contrast, the Vision Transformer (ViT)-Hybrid architecture integrates CNN-based feature extraction with transformer self-attention

mechanisms to capture global pixel dependencies more effectively [20]. These architectures were selected to systematically analyze the trade-offs between classification accuracy and computational efficiency in avocado leaf image classification [14, 15]. The results demonstrate that DCNN-based models can accurately classify avocado varieties using leaf morphological features, effectively capturing subtle visual differences that are challenging for human observation [32]. Moreover, the findings highlight the potential of image-based DCNN approaches as practical and scalable solutions for plant identification in modern agricultural applications.

3.1. Science Aspect

The **Science** aspect of this research focuses on the DCNN used for avocado leaf classification based on the basic principles of avocado plant morphology and genetic diversity. Avocado leaf morphology forms phenotypic variations resulting from a combination of environmental and genetic factors [33, 35, 36]. Generally, each avocado variety has unique leaf shape characteristics. The Hass avocado variety produces elongated elliptical fruit, while the Pluwang and Clara varieties produce wide oval fruit. In addition to genetic factors, morphological plasticity also plays an important role. The ratio of leaf length to width is often an important differentiating factor[34]. This is indicated by some varieties having slimmer proportions and others having wider leaves [37, 38]. In addition, leaf vein patterns also play an important role as morphological markers. Clear and contrasting patterns are shown by some avocado varieties, while others display subtle patterns that are difficult to detect with the naked eye [35, 39, 40]. DCNN can identify these differences through a feature extraction process. This process works by highlighting fine lines of leaf veins that are often missed in manual observations.

Figure 10. Augmentation and image extraction of avocado leaf

The dataset consisted of 7000 images of avocado leaves representing seven distinct varieties. The Representative examples of augmented avocado leaf images across seven classes show on Figure 10. Data augmentation was applied to increase sample diversity and prevent overfitting, involving random rotation ($\pm 20°$), horizontal and vertical flips, brightness adjustment (0.8–1.2×), and slight zoom perturbations. Furthermore, Figure 10 illustrates representative samples after preprocessing, showing sufficient intra-class diversity and inter-class separability. The next stage was carried out by converting digital images of avocado leaves into RGB images by identifying the dataset and classifying it based on the characteristics of each dataset. Additionally, Figure 10 shows an example of an avocado leaf sample that has been converted into an RGB image and then processed using the DCNNs algorithms to produce avocado leaf identification results. This is done by using a filter of a certain size that is shifted across the entire image. In the convolution process, a "*dot*" operation is performed between the input values and the filter to produce an activation map or feature map. This extraction process is performed on all RGB spectra.

The horticulture experts confirmed that the regions emphasized by the Grad-CAM visualizations corresponded to morphologically relevant features commonly used to distinguish avocado varieties, particularly vein structure and leaf margin characteristics. This agreement supports the biological validity of the learned representations. The alignment between XAI-identified regions and expert-recognized morphological traits enhances the practical relevance of the proposed model. Such transparency is essential for building trust in AI-assisted agricultural applications and supports the model's potential adoption in real-world horticultural decision-making. Although extensive field testing was beyond the scope of this study, the successful implementation of the proposed model within an Android application demonstrates its potential for real-world agricultural deployment. As illustrated in Figure 11, the Grad-CAM visualizations consistently highlight diagnostically relevant leaf regions across different avocado varieties, providing visual confirmation of the alignment between model attention and expert-identified morphological features.

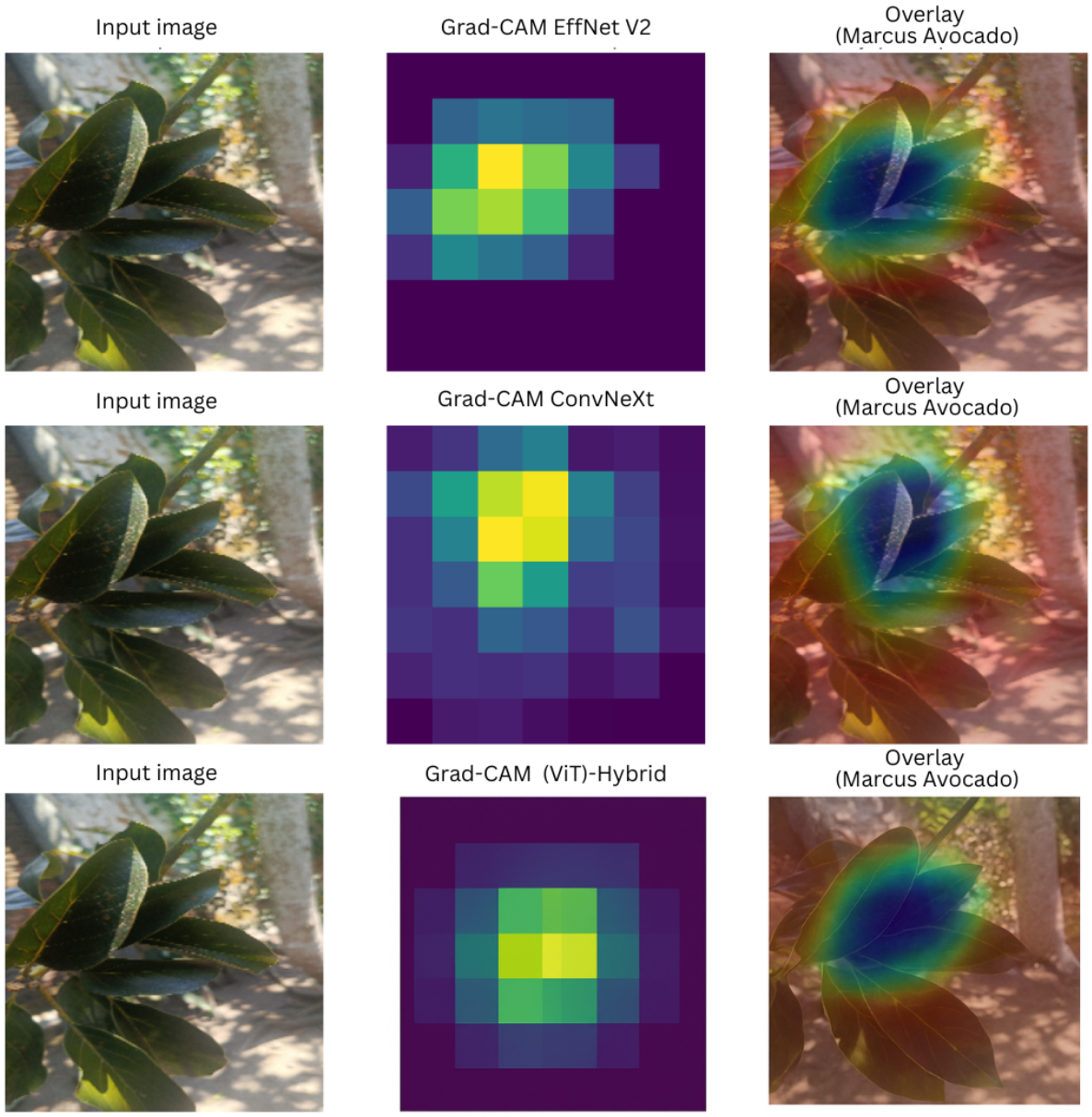

Figure 11. Grad-CAM Visualisation as Explainable Ai

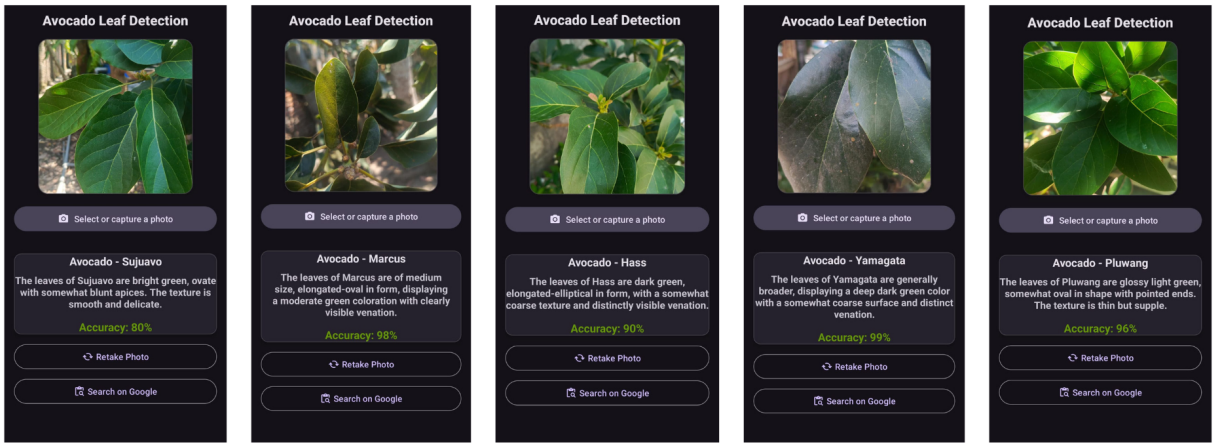

Figure 12. Android Application for Agricultural Deployment

To enhance practical usability, informal qualitative feedback was collected from farmers and horticulture practitioners during prototype testing of the Android application as shown in Figure 12. The feedback emphasized the importance of clear image acquisition guidelines, appropriate lighting conditions, and a simple interface for presenting classification results. These insights were used to refine the application workflow and will guide future improvements. To assess the generalization capability of the proposed models beyond controlled experimental settings, an independent set of field images was employed for external validation. These images were captured using a smartphone-based application under real-world conditions, including varying illumination, complex backgrounds, and natural leaf occlusions, and were entirely excluded from the training and cross-validation process. Although a modest decrease in accuracy was observed compared to the controlled dataset, the models maintained strong performance, demonstrating robustness and suitability for real-world deployment. While multi-modal imaging systems, such as hyperspectral cameras, can potentially enhance classification accuracy by capturing detailed physiological and biochemical leaf traits, they entail high costs, hardware complexity, and substantial computational overhead, limiting their practicality for large-scale or smallholder agricultural use. In contrast, RGB-based approaches combined with low-cost sensing solutions, such as smartphone-assisted chlorophyll estimation or simple field measurements, offer a more balanced trade-off between performance and practicality, supporting the adoption of RGB-driven deep learning models as accessible and scalable tools for avocado variety identification.

3.2. Engineering and Mathematics Aspect

The **Engineering** aspects carried out are part of the engineering design process, This stage begins with resizing, which is the process of changing the image size to a standard resolution that is compatible with the DCNN architecture. Resizing not only reduces computational complexity, but also ensures that each image has a uniform spatial representation so that the convolution kernel can extract features with high consistency [50]. After the image size has been standardised, the next step is normalisation, which is the process of changing the pixel intensity scale from a range of [0–255] to a scale of [0–1] or [-1–1] [51, 52].

3.2.1. EfficientNetV2

EfficientNet uses Fused-MBConv with nine parameters, including input image, kernel size, input channels, output channels, stride, padding, activation, batch normalisation, and pooling. To illustrate the mechanism, consider a 3×3 patch from a red channel:

$$P_{red} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 69 & 72 \\ 0 & 118 & 74 \end{bmatrix} * \begin{bmatrix} 2 & -2 & 1 \\ 1 & 1 & -1 \\ 1 & 0 & -1 \end{bmatrix}$$
$$= (0 * 2) + (0 * -2) + ... + (74 * (-1))$$
$$P_{red} = -77$$

then from a green channel:

$$P_{green} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 83 & 87 \\ 0 & 119 & 84 \end{bmatrix} * \begin{bmatrix} 2 & -2 & 1 \\ 1 & 1 & -1 \\ 1 & 0 & -1 \end{bmatrix}$$
$$= (0 * (2)) + (0 * -2) + ... + (84 * -1)$$
$$P_{green} = -88$$

then from a blue channel:

$$P_{blue} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 58 & 66 \\ 0 & 92 & 59 \end{bmatrix} * \begin{bmatrix} 2 & -2 & 1 \\ 1 & 1 & -1 \\ 1 & 0 & -1 \end{bmatrix}$$
$$= (0 * 2) + (0 * -2) + ... + (59 * -1)$$
$$P_{blue} = -67$$

The convolution (without bias) produces the output value at the center position as:

$$y = \sum_{i=1}^{3} \sum_{j=1}^{3} K_{i,j} \cdot P_{i,j}$$

The resulting future map values are as follows:

$$y = P_{red} + P_{green} + P_{blue}$$
$$= -77 + (-88) + (-67)$$
$$y = -231$$

If we have a bias term b ($b = 1$), then the final output before Batch Normalization or activation is:

$$y_{\text{bias}} = -231 + 1 = -230.$$

Convolution 3×3 (Fused-MBConv Basic mathematical equation of 2D convolution:

$$Y(i, j, k) = \sum_{m=1}^{K} \sum_{n=1}^{K} \sum_{c=1}^{C_{\text{in}}} W_{m,n,c,k} \; X_{(i+m-1),(j+n-1),c} \; + \; b_k$$

$$P_{\text{total}} = K^2 \times C_{\text{in}} \times C_{\text{out}} + C_{\text{out}}$$

$$P_{\text{total}} = 3^2 \times 3 \times 32 + 32$$
$$= 9 \times 3 \times 32 + 32$$
$$= 864 + 32$$
$$= 896 \quad \text{parameters}$$

Global Average Pooling + Dense Layer GAP converts a tensor of size $7 \times 7 \times 1280$ into a vector of 1×1280. Dense layer (for example, 7 leaf classes):

$$P_{\text{dense}} = 1280 \times 7 + 7 = 8967$$

The total number of parameters and floating-point operations in EfficientnetV2 block are summarized in Table 6.

Table 6. Parameter Count and Computational Cost (FLOPs)

Component	**Parameters**	**FLOPs**
Conv 3×3	896	43.36 M
BatchNorm	64	0.20 M
1×1 Conv	2,048	102.8 M
GAP + Dense	8,967	0.01 M
Total (1 Block)	**11,975**	**$\approx$ 146 M**

3.2.2. ConvNeXt

The ConvNeXt architecture modernizes the classical ResNet structure by integrating depthwise separable convolutions, layer normalization, and feed-forward expansions inspired by Vision Transformers (ViT). Let the input image be denoted by:

$$X_0 \in \mathbb{R}^{224\times224\times3},$$

Depthwise Convolution + LayerNorm Within each ConvNeXt block, a depthwise convolution captures spatial patterns:

$$Y_{i,j,c} = \sum_{m=-3}^{3}\sum_{n=-3}^{3} X_{1(i+m,j+n,c)} \cdot K_{m,n,c},$$

Channels are projected back to their original size:

$$P_{i,j,c} = \sum_{k=1}^{384} A_{i,j,k} \cdot W_{k,c}^{(2)} + b_c^{(2)},$$

This preserves low-level spatial cues while enriching contextual representation.

Each ConvNeXt stage reduces spatial resolution by half and doubles the channel depth:

$$\begin{aligned} X_2 &\in \mathbb{R}^{56\times56\times96}, \\ X_3 &= \text{Downsample}(X_2) \in \mathbb{R}^{28\times28\times192}, \\ X_4 &= \text{Downsample}(X_3) \in \mathbb{R}^{14\times14\times384}, \\ X_5 &= \text{Downsample}(X_4) \in \mathbb{R}^{7\times7\times768}. \end{aligned}$$

The ConvNeXt architecture was performed using a $3 \times 3 \times 3$ RGB image input including depthwise convolution, layer normalization, expansion and projection via 1×1 convolutions, activation with GELU, and residual addition. The input tensor $X \in \mathbb{R}^{3\times3\times3}$ represents an RGB image with values:

$$X^{(red)} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 69 & 72 \\ 0 & 118 & 74 \end{bmatrix}, \quad X^{(green)} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 83 & 87 \\ 0 & 119 & 84 \end{bmatrix}, \quad X^{(blue)} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 58 & 66 \\ 0 & 92 & 59 \end{bmatrix}.$$

Each color channel is processed by a separate depthwise kernel $K^{(c)} \in \mathbb{R}^{3\times3}$:

$$K^{(1)} = \begin{bmatrix} 2 & -2 & 1 \\ 1 & 1 & -1 \\ 1 & 0 & -1 \end{bmatrix}, \quad K^{(2)} = \begin{bmatrix} 2 & -2 & 1 \\ 1 & 1 & -1 \\ 1 & 0 & -1 \end{bmatrix}, \quad K^{(3)} = \begin{bmatrix} 2 & -2 & 1 \\ 1 & 1 & -1 \\ 1 & 0 & -1 \end{bmatrix}.$$

The depthwise convolution output for each channel at the center position $(2, 2)$ is computed as:

$$y^{(c)} = \sum_{i=1}^{3}\sum_{j=1}^{3} X_{ij}^{(c)} K_{ij}^{(c)}.$$

The numerical results are:

$$y^{(1)} = -77, \quad y^{(2)} = -88, \quad y^{(3)} = -67.$$

Thus, the depthwise output vector at the center location is:

$$Y_{\text{dw}}(2,2) = [\,-77,\ -88,\ -67\,]^T.$$

Substituting $[-77, -88, -67]$, we obtain:

$$\mu = -77.3333, \quad \sigma = 8.5741,$$

$$\hat{Y}(2,2) = [0.0389,\ -1.2457,\ 1.2067].$$

The normalized vector is projected to a higher-dimensional feature space (expansion ratio $r = 2$):

$$Z = \hat{Y} \times W^{(1)},$$

where

$$W^{(1)} = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 \end{bmatrix}.$$

The resulting expanded vector is:

$$Z = [1.2456,\ -0.0390,\ 0.0389,\ 1.2457,\ 1.2456,\ 0.0390].$$

The total number of parameters (P) and floating-point operations (F) in ConvNeXt block are summarized in Table 7.

Table 7. Mathematical Summary of ConvNeXt Block ($3 \times 3 \times 3$ input)[**?**, **?**]

Operation	**Parameters**	**FLOPs**
Depthwise Conv (3×3)	$9 \times 3 = 27$	$3 \times 3 \times 9 \times 3 = 243$
Expansion (1×1, $3 \to 6$)	$3 \times 6 = 18$	$3 \times 3 \times 3 \times 6 = 162$
Projection (1×1, $6 \to 3$)	$6 \times 3 = 18$	$3 \times 3 \times 6 \times 3 = 162$
LayerNorm + GELU	6	~ 675
Total	**69**	$\approx 1.24 \times 10^3$

3.2.3. ViT-Hybrid CNN

The input image $I \in \mathbb{R}^{H \times W \times C_{in}}$ (where $H = W = 224$, $C_{in} = 3$ for RGB channels) is first processed by a convolutional stem composed of one or more convolutional layers with kernel size $K \times K$, stride s, and output channel C_{out}. Each convolutional layer performs the operation:

$$F_{i,j,c_{out}} = \sum_{u=1}^{K} \sum_{v=1}^{K} \sum_{c_{in}=1}^{C_{in}} W_{u,v,c_{in},c_{out}} \cdot I_{i+u,j+v,c_{in}} + b_{c_{out}},$$

The total number of parameters and floating-point operations (FLOPs) for one convolutional layer are:

$$P_{\text{conv}} = K^2 \cdot C_{in} \cdot C_{out}, \qquad \text{FLOPs}_{\text{conv}} = H'W'K^2C_{in}C_{out}.$$

The number of parameters and FLOPs for MHA are:

$$P_{\text{MHA}} = 4D^2, \qquad \text{FLOPs}_{\text{MHA}} = 4ND^2 + 4N^2D.$$

For $D = 384$, $N = 196$, and $h = 6$ heads, the MHA contributes:

$$P_{\text{MHA}} = 4(384)^2 = 589{,}824, \quad \text{FLOPs}_{\text{MHA}} = 4(196)(384)^2 + 4(196)^2(384) \approx 2.56\text{G FLOPs}.$$

The Feed-Forward Network (FFN) applies two dense layers with an expansion ratio $r = 4$:

$$\text{FFN}(x) = \sigma(xW_1 + b_1)W_2 + b_2,$$

For $r = 4$ and $D = 384$, $P_{\text{FFN}} = 2(384)^2(4) = 1.18\text{M}$ parameters, and $\text{FLOPs}_{\text{FFN}} = 2(196)(384)^2(4) \approx 0.46\text{G}$ FLOPs.

Combining all components, the total parameters and FLOPs for the ViT-Hybrid CNN are:

$$P_{\text{total}} = P_{\text{stem}} + P_{\text{proj}} + L(4D^2 + 2D^2 r) + P_{\text{cls}},$$

$$\text{FLOPs}_{\text{total}} = \text{FLOPs}_{\text{stem}} + NCD + L(4ND^2 + 4N^2 D + 2ND^2 r).$$

For a configuration with $L = 12$ layers, $D = 384$, $r = 4$, and $N = 196$, the total FLOPs reach approximately 35.5 GFLOPs and $P_{\text{total}} \approx 13.2$ M parameters. The total number of parameters and floating-point operations in ViT-Hybrid CNN are summarized in Table 8

Table 8. Summary of Mathematical Formulation for ViT-Hybrid CNN

Stage	**Operation**	**Parameters**	**FLOPs**	**Example (D=384, N=196)**
Conv Stem	$K^2 C_{in} C_{out}$	1.7k	87M	Local feature extraction
Tokenisation	$C_f D$	147k	28.9M	Linear patch embedding
Self-Attention	$4D^2$	589k	2.56G	Global context modelling
Feed-Forward	$2D^2 r$	1.18M	0.46G	Nonlinear transformation
Classifier	DN_c	3k	0.001G	Output mapping
Total (per layer)	–	1.77M	3.02G	–
Total (L=12)	–	13.2M	35.5G	–

In the convolutional stem of the ViT-Hybrid CNN, the input image $I \in \mathbb{R}^{H \times W \times C_{in}}$ with three channels (RGB) is processed by a convolutional kernel $W \in \mathbb{R}^{K \times K \times C_{in} \times C_{out}}$ to generate the feature map F. Each output activation $F_{i,j,c_{out}}$ is computed according to the standard convolution operation:

$$F_{i,j,c_{out}} = \sum_{u=1}^{K} \sum_{v=1}^{K} \sum_{c_{in}=1}^{C_{in}} W_{u,v,c_{in},c_{out}} \cdot I_{i+u,j+v,c_{in}} + b_{c_{out}},$$

where $K = 3$ denotes the kernel size, $C_{in} = 3$ represents the RGB input channels, and $b_{c_{out}}$ is the bias term.

For this calculation, the input image channels and kernels are defined as follows:

$$I^{(\text{red})} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 69 & 72 \\ 0 & 118 & 74 \end{bmatrix}, \quad I^{(\text{green})} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 83 & 87 \\ 0 & 119 & 84 \end{bmatrix}, \quad I^{(\text{blue})} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 58 & 66 \\ 0 & 92 & 59 \end{bmatrix},$$

$$W^{(1)} = W^{(2)} = W^{(3)} = \begin{bmatrix} 2 & -2 & 1 \\ 1 & 1 & -1 \\ 1 & 0 & -1 \end{bmatrix}, \quad b_{c_{out}} = 1.$$

Aggregating across the RGB channels and adding the bias term yields:

$$F_{i,j,c_{out}} = (-77) + (-88) + (-67) + 1 = -231.$$

Therefore, for the given kernel and input configuration, the convolutional stem produces an activation value of:

$$\boxed{F_{i,j,c_{out}} = -231.}$$

These results demonstrate how much the accumulated local pixel intensity per channel is modulated by kernel weights with the addition of bias. In deeper layers of the ViT-Hybrid CNN, multiple kernels (C_{out}) extract various spatial features simultaneously, which are then flattened and projected into a token embedding for transformer processing.

3.3. Technology Aspect

The technological aspects of this study demonstrate the testing of three artificial neural network architectures, namely EfficientNetV2, ConvNeXt, and Vision Transformer (ViT)-Hybrid. The EfficientNetV2 model is able to capture local and global image features simultaneously so that it can classify various plant leaves with high accuracy

[41, 42]. On the other hand, ConvNeXt, which is a modern CNN, can also improve the representation of spatial and texture features, making it effective for agricultural image analysis [43]. Previous research has also been conducted on avocado plants. EfficientNet was able to achieve an accuracy of more than 90% in detecting fruit ripeness [44] while ViT can outperform CNN in terms of the process of learning global spatial relationships for leaf and fruit features [45]. EfficientNetV2 and ConvNeXt also show extraordinary capabilities in distinguishing vein patterns and textures in various plant species [46, 47]. In addition, corn and mulberry plants have been studied using a hybrid CNN–ViT architecture with better accuracy results. This occurs because the hybrid CNN-ViT combines the local power of CNN with the global context modeling of Transformer [48, 49]. EfficientNetV2 has the lowest complexity of 1.34 GFLOPs and 7.1 M parameters. In contrast, the hybrid CNN-ViT requires higher computation due to the presence of the Transformer encoder layer. However, its accuracy gain justifies the increased cost, especially for high-performance computing environments. The computational complexity from each architectures show on Table 9 below.

Table 9. Computational complexity of the evaluated models.

Model	**Parameters (M)**	**FLOPs (GFLOPs)**
EfficientNetV2	7.10	1.34
ConvNeXt	28.30	4.50
ViT-Hybrid CNN	33.25	4.97

Table 10. Comparison of model performance on the avocado leaf dataset.

Model	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1-Score (%)**
EfficientNetV2	**98.71**	**98.43**	**95.16**	**97.57**
ConvNeXt	97.92	96.22	94.31	95.15
ViT-Hybrid CNN	97.24	95.27	96.33	95.42

Table 11. Comparative Performance and Computational Efficiency of Deep Learning Models

Model	**Accuracy (%)**	**Params (M)**	**Model Size (MB)**	**Inference Time (ms)**
EfficientNetV2	**98.71**	21.5	82	12.3
ConvNeXt	97.92	28.6	110	18.9
ViT-Hybrid	97.24	34.1	132	24.5
MobileNetV3	95.40	**5.4**	**21**	**6.1**
ShuffleNet	94.88	**3.1**	**13**	**4.9**
EffNetV2-Lite	96.15	7.2	28	7.4

Table 12. Comparison of Model Performance Between Controlled and Real-World Evaluation Settings

Evaluation Setting	**Accuracy (%)**	**Avg. Inference Time (ms)**
Controlled dataset	98.7	12.3
Real-world images	95.8	13.4

Table 13. Paired t-test Results Comparing Classification Accuracy Across Models

Model A	Model B	Mean Acc A (%)	Mean Acc B (%)	p-value	Significant
EfficientNetV2	ConvNeXt	98.71	97.90	<0.001	Yes
EfficientNetV2	ViT-Hybrid	98.71	97.23	<0.001	Yes
EfficientNetV2	MobileNetV3	98.71	95.41	<0.001	Yes
EfficientNetV2	ShuffleNet	98.71	94.79	<0.001	Yes
EfficientNetV2	EfficientNetV2-Lite	98.71	96.10	<0.001	Yes
ConvNeXt	ViT-Hybrid	97.90	97.23	<0.001	Yes
ConvNeXt	MobileNetV3	97.90	95.41	<0.001	Yes
ConvNeXt	ShuffleNet	97.90	94.79	<0.001	Yes
ConvNeXt	EfficientNetV2-Lite	97.90	96.10	<0.001	Yes
ViT-Hybrid	MobileNetV3	97.23	95.41	<0.001	Yes
ViT-Hybrid	ShuffleNet	97.23	94.79	<0.001	Yes
ViT-Hybrid	EfficientNetV2-Lite	97.23	96.10	<0.001	Yes
MobileNetV3	ShuffleNet	95.41	94.79	<0.001	Yes
MobileNetV3	EfficientNetV2-Lite	95.41	96.10	<0.001	Yes
ShuffleNet	EfficientNetV2-Lite	94.79	96.10	<0.001	Yes

Table 10 summarizes the performance of the evaluated architectures. EfficientNetV2 achieved the highest accuracy and F1-score while maintaining a favorable balance between accuracy and computational efficiency, confirming its effectiveness in capturing both fine-grained morphological details and broader contextual information. ConvNeXt delivered competitive performance with moderate computational cost, whereas the ViT-Hybrid CNN provided stable results by combining CNN-based feature extraction with transformer-based global dependency modeling. The superior performance of EfficientNetV2 is attributed to the use of Fused-MBConv blocks, small 3×3 convolutional kernels that effectively capture leaf venation patterns, and compound scaling that optimizes accuracy relative to FLOPs. Under unconstrained field conditions, the model maintained robust performance with only a modest accuracy reduction, while inference time remained stable, supporting real-time applicability.

To assess deployment feasibility, lightweight models including MobileNetV3, ShuffleNet, and EfficientNetV2-Lite were evaluated see Table 11. Although these architectures exhibited slightly lower accuracy, they offered substantial reductions in model size and inference time. EfficientNetV2-Lite reduced model size by approximately 66% with a 2.6% accuracy decrease, while MobileNetV3 and ShuffleNet achieved model size reductions exceeding 74% and 84%, respectively, at the cost of modest accuracy loss. These results highlight a clear trade-off between classification accuracy and deployment efficiency, indicating that the proposed approach remains practical and effective for real-world agricultural and edge-based applications. Table 12 presents a comparative evaluation of model performance under controlled and real-world testing conditions. The results indicate that the proposed model maintains a consistent level of performance when transitioning from controlled datasets to images acquired in real-world environments. Although minor variations are observed due to increased environmental complexity, the overall stability of the model demonstrates its capacity to generalize beyond idealized experimental settings. Additionally, the inference process remains efficient across both evaluation scenarios, supporting the feasibility of deploying the proposed system in practical, real-world agricultural applications. On the other hand, the paired t-test analysis confirms that EfficientNetV2 significantly outperforms ConvNeXt, ViT-Hybrid, and lightweight models ($p < 0.05$), validating that the observed performance gains are statistically meaningful.

In addition to statistical significance testing, model robustness was further supported by narrow 95% confidence intervals obtained from the 10-fold cross-validation. The consistently small confidence ranges for accuracy, precision, and recall indicate low variance across folds, confirming stable and reliable model performance under different data partitions. As reported in Table13, all paired model comparisons show statistically significant differences ($p < 0.05$), demonstrating that the observed performance gaps are not attributable to random chance. Notably, EfficientNetV2 achieved a mean accuracy of 98.71% with a 95% confidence interval of [98.45%, 98.97%], further reinforcing the reproducibility and robustness of the proposed approach.

3.4. Mathematics Aspect

The **Mathematics** aspect represents deep convolutional neural networks (DCNNs) as the core machine learning mechanism for avocado leaf morphology classification, where feature extraction is governed by structured convolution operations that compute dot products between learnable kernels and local image regions [53]. The models were trained for 100 epochs with a batch size of 32, an initial learning rate of 1×10^{-4}, and the Adam optimizer, incorporating early stopping and learning rate reduction on plateau to ensure stable convergence. The results reveal a clear trade-off between classification accuracy and computational efficiency: EfficientNetV2 achieves the highest accuracy for detailed leaf analysis, while lightweight architectures offer practical alternatives for real-time, resource-constrained edge deployment. Figure 13 show the validation accuracy across 10-fold cross validation and loss for all evaluated models. EfficientNetV2 consistently achieves the highest accuracy across folds, followed by ConvNeXt and ViT-Hybrid CNN. Lightweight models (MobileNetV3, ShuffleNet, and EfficientNetV2-Lite) exhibit slightly lower but stable accuracy, highlighting the trade-off between classification performance and deployment efficiency. Validation loss across 10-fold cross-validation for all evaluated models. EfficientNetV2 consistently achieves the lowest loss across folds, indicating superior optimization stability. Lightweight models exhibit higher but stable loss values, reflecting the trade-off between computational efficiency and classification performance.

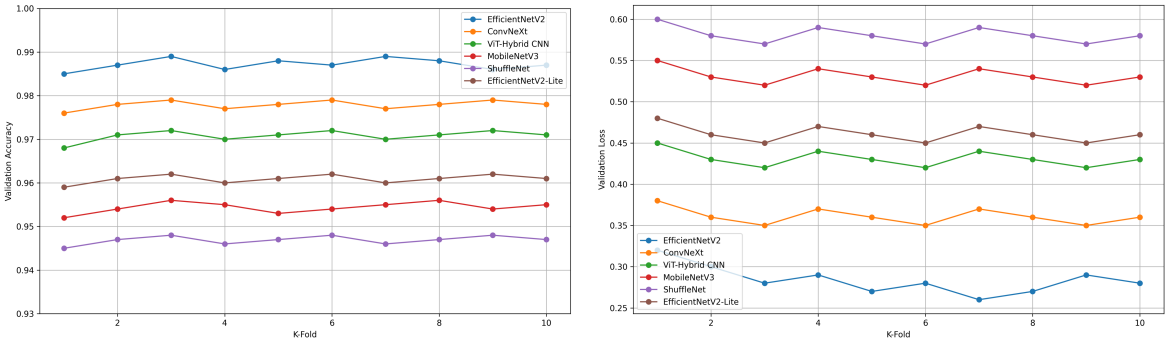

Figure 13. Accuracy comparison of all architectures

The confusion matrices for all architectures are shown in Figure 14. The confusion matrices provide a detailed class-wise evaluation of the six evaluated architectures, revealing clear differences in classification consistency and error distribution across avocado varieties. EfficientNetV2 demonstrates the strongest overall performance, with highly dominant diagonal values across all classes, indicating precise discrimination among avocado varieties. Misclassifications are minimal and primarily occur between morphologically similar classes, such as Marcus and Aligator, suggesting that remaining errors are attributable to subtle inter-class similarities rather than model instability. ConvNeXt achieves competitive performance but exhibits increased off-diagonal confusion compared to EfficientNetV2, particularly between Hass, Marcus, and Clara. This pattern indicates a reduced ability to capture fine-grained leaf venation and margin characteristics under similar visual conditions. ViT-Hybrid CNN shows a more balanced yet diffused confusion pattern, with higher misclassification rates across multiple classes. While the hybrid architecture benefits from global attention, its performance suggests that transformer-based representations alone are less effective than convolutional inductive biases for fine-grained leaf morphology classification. Among the lightweight models, EfficientNetV2-Lite consistently outperforms MobileNetV3 and ShuffleNet, maintaining strong diagonal dominance and relatively low inter-class confusion. This indicates that compound scaling and efficient feature reuse enable EfficientNetV2-Lite to preserve discriminative capacity despite reduced computational complexity. MobileNetV3 exhibits moderate confusion across visually overlapping classes, reflecting the trade-off between parameter efficiency and feature expressiveness. In contrast, ShuffleNet presents the highest level of misclassification dispersion, with widespread off-diagonal entries, highlighting its limited representational capacity for subtle morphological variations. Overall, the confusion matrix analysis confirms that EfficientNetV2 and EfficientNetV2-Lite offer the most reliable class-wise performance, achieving superior balance between accuracy and robustness. The observed error patterns across all architectures consistently align with

morphological similarities among avocado varieties, reinforcing the biological plausibility of the model predictions. These results further support the suitability of EfficientNet-based architectures for both high-accuracy classification and resource-efficient real-world deployment.

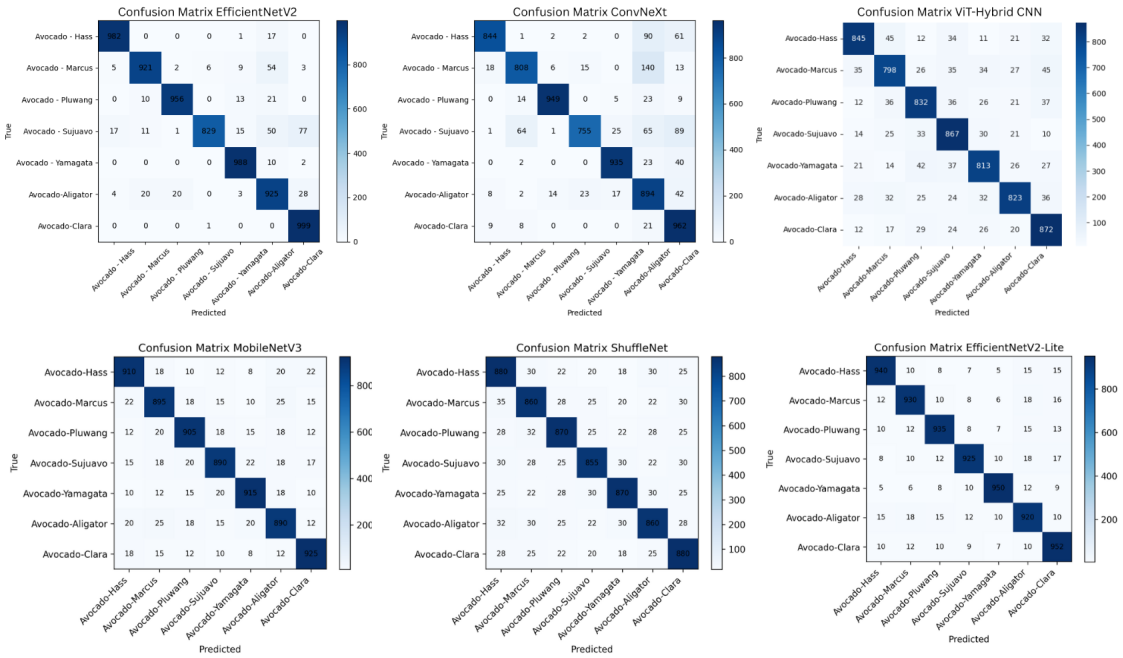

Figure 14. Confusion matrix analysis from all architectures

Comparative evaluation shows that each architecture has its own advantages as shown in Table 14. The EfficientNetV2 architecture is able to offer outstanding parameter efficiency through composite scaling and integrated convolution blocks, making it ideal for lightweight applications. In contrast, the ConvNeXt architecture modernizes CNN design by combining transformer-inspired normalization and large-kernel depthwise convolution to achieve a balance between accuracy and efficiency. The final architecture, the ViT-Hybrid CNN, is able to deliver optimal performance by combining convolutional feature extraction with global attention to improve the representation of long-range dependencies. These findings corroborate recent studies on high-resolution image classification [24, 25], which confirmed the efficiency and minimal complexity of the EfficientNetV2 architecture.

Table 14. Computational Complexity Comparison of Deep Learning Architectures

Architecture	**Approx. FLOPs (Millions)**	**Core Advantage**
EfficientNetV2	8–10	High efficiency achieved through compound scaling and Fused-MBConv design
ConvNeXt	12–15	Improved training stability and enhanced multi-scale feature representation
ViT-Hybrid CNN	20–25	Combination of local CNN feature extraction with global Transformer attention
MobileNetV3	3–5	Extremely lightweight architecture using depthwise separable convolutions and squeeze-and-excitation blocks, optimized for low-latency mobile inference
ShuffleNet	2–3	Channel shuffling mechanism enables efficient information flow with minimal computational cost, suitable for ultra-low-resource edge deployment
EfficientNetV2-Lite	4–6	Simplified compound scaling strategy that balances accuracy and efficiency while maintaining strong performance on resource-constrained devices

Although encouraging results were obtained in this study, several limitations must be acknowledged in its implementation. First, although the dataset has been expanded to include environmental variability and external regional evaluations, the total number of samples used remains moderate compared to the large-scale benchmark datasets commonly used in deep learning research. As a result, further research could be developed by expanding the dataset to cover a wider geographical area with seasonal variations. Second, although multimodal integration showed better performance, data availability is still limited in the context of small-scale agriculture. Therefore, continuous exploration is needed using cost-efficient camera sensors for more practical applications.

4. Conclusion

This study presented a STEM-based deep learning framework for avocado (*Persea americana* Mill.) leaf variety classification using advanced DCNN architectures. By integrating systematic field data acquisition, robust statistical evaluation, and explainable artificial intelligence (XAI), the proposed approach addressed the challenges of fine-grained morphological similarity and environmental variability in real-world agricultural settings. A comprehensive dataset consisting of seven avocado varieties was evaluated using 10-fold cross-validation to ensure reliable performance estimation. Among the evaluated architectures, EfficientNetV2 consistently achieved superior classification performance, attaining a mean accuracy of 98.71% with narrow 95% confidence intervals, and significantly outperforming ConvNeXt and ViT-Hybrid CNN models based on paired t-test analysis ($p < 0.05$). Explainability analysis using Grad-CAM demonstrated that the models focused on biologically meaningful leaf regions, such as venation patterns and leaf margins, supporting the interpretability and practical relevance of the classification results. Additional experiments under unconstrained field conditions revealed only a modest reduction in accuracy, indicating strong model robustness and suitability for real-world deployment. Furthermore, the inclusion of lightweight architectures highlighted a clear trade-off between classification accuracy and deployment feasibility. While EfficientNetV2 provided the highest accuracy, lightweight variants such as EfficientNetV2-Lite, MobileNetV3, and ShuffleNet offered substantially reduced model size and faster inference, making them attractive for edge-based and smartphone applications. These findings demonstrate that RGB-based DCNN models can deliver accurate, interpretable, and deployable solutions for avocado variety identification using low-cost imaging devices. Future work will focus on expanding external validation across broader geographic regions, incorporating domain expert-driven feature analysis, and exploring practical multi-modal extensions that balance performance gains with deployment cost. Overall, this study contributes a robust and scalable framework that supports precision agriculture and digital plant identification in real-world scenarios.

Acknowledgement

The authors gratefully acknowledge the support provided by Universitas PGRI Argopuro Jember. The authors also extend their sincere appreciation to the PUI-PT CGANT research team at Universitas Jember for their valuable contributions, collaboration, and support, which made this research possible.

REFERENCES

1. Garbero, A., & Jäckering, L. (2021). The potential of agricultural programs for improving food security: A multi-country perspective. Global Food Security, 29, 100529. https://doi.org/10.1016/j.gfs.2021.100529
2. Paniza, H. M. (2024). Challenges in Plant Breeding Under Climate Change: A Review. Plant Quarantine Challenges under Climate Change Anxiety, 533–556. https://doi.org/10.1007/978-3-031-56011-8_17
3. Kuhnlein, H. V., & Receveur, O. (1996). Dietary change and traditional food systems of Indigenous peoples. Annual Review of Nutrition, 16, 417–442. https://doi.org/10.1146/annurev.nu.16.070196.002221
4. Dreher, M. L., & Davenport, A. J. (2013). Hass avocado composition and potential health effects. Critical Reviews in Food Science and Nutrition, 53(7), 738–750. https://doi.org/10.1080/10408398.2011.556759
5. Slavin, J. L., & Lloyd, B. (2012). Health benefits of fruits and vegetables. Advances in Nutrition, 3(4), 506–516. https://doi.org/10.3945/an.112.002154
6. Donetti, M., & Terry, L. A. (2014). Biochemical markers defining growing area and ripening stage of imported avocado fruit cv. Hass. Journal of Food Composition and Analysis, 34(1), 90–98. https://doi.org/10.1016/j.jfca.2014.01.008
7. Saucedo-Carabez, J. R., Teliz-Ortiz, D., Ochoa-Ascencio, S., Ochoa-Martinez, D., Vallejo-Perez, M. R., & Beltran-Pena, H. (2015). Effect of Avocado sunblotch viroid (ASBVd) on the Postharvest Quality of Avocado Fruits from Mexico. Journal of Agricultural Science, 7(9). https://doi.org/10.5539/jas.v7n9p85
8. Too, E. C., Yujian, L., Njuki, S., & Yingchun, L. (2019). A comparative study of fine-tuning deep learning models for plant disease identification. Computers and Electronics in Agriculture, 161, 272–279. https://doi.org/10.1016/j.compag.2018.03.032

9. Tiwari, S., Nigam, S., & Singh, V. (2022). Deep convolutional neural networks with transfer learning and data augmentation for plant disease detection. Ecological Informatics, 67, 101465. https://doi.org/10.1016/j.ecoinf.2021.101465
10. Ferentinos, K. P. (2018). Deep learning models for plant disease detection and diagnosis. Computers and Electronics in Agriculture, 145, 311–318. https://doi.org/10.1016/j.compag.2018.01.009
11. Barbedo, J. G. A. (2013). Digital image processing techniques for detecting, quantifying and classifying plant diseases. SpringerPlus, 2(1), 660. https://doi.org/10.1186/2193-1801-2-660
12. Gill, T., & Yadav, S. (2021). STEM-based learning framework for scientific and technological skill development in agriculture. Journal of Educational Technology & Society, 24(1), 203–214. https://www.jstor.org/stable/26910199
13. Singh, V., & Misra, A. K. (2017). Detection of plant leaf diseases using image segmentation and soft computing techniques. Information Processing in Agriculture, 4(1), 41–49. https://doi.org/10.1016/j.inpa.2016.10.005
14. Mohanty, S. P., Hughes, D. P., & Salathé, M. (2016). Using deep learning for image-based plant disease detection. Frontiers in Plant Science, 7, 1419. https://doi.org/10.3389/fpls.2016.01419
15. Tan, M., & Le, Q. V. (2021). Efficientnetv2: smaller models and faster training. International Conference on Machine Learning, 10096-10106. 10.48550/arXiv.2104.00298
16. Zhao, H., Zhang, X., Gao, Y., Wang, L., Xiao, L., Liu, S., Huang, B., & Li, Z. (2024). Diagnostic performance of efficientnetv2-s method for staging liver fibrosis based on multiparametric MRI. Heliyon, 10(15), 1-11. https://doi.org/10.1016/j.heliyon.2024.e35115
17. Hassan, E., & Ghadiri, H. (2025). Advancing brain tumor classification: A robust framework using efficientnetv2 transfer learning and statistical analysis. Computers in Biology and Medicine, 185, 109542. https://doi.org/10.1016/j.compbiomed.2024.109542
18. Abioye, O. A., Evwiekpaefe, A. E., & Olalekan, A. J. (2024). Performance evaluation of efficientnetv2 models on the classification of histopathological benign breast cancer images. Science Journal of University of Zakho, 12(2), 208-214. https://doi.org/10.25271/sjuoz.2024.12.2.1261
19. Guo, B., Qiao, Z., Zhang, N., Wang, Y., Wu, F., & Peng, Q. (2024). Attention-based convneXt with a parallel multiscale dilated convolution residual module for fault diagnosis of rotating machinery. Expert Systems with Applications, 249, 123764. https://doi.org/10.1016/j.eswa.2024.123764
20. Wang, G., Chen, H., Chen, L., Zhuang, Y., Zhang, S., Zhang, T., Dong, H., & Gao, P. (2023). P^2fevit: plug-and-play cnn feature embedded hybrid vision transformer for remote sensing image classification. Remote Sensing, 15(7), 1773. https://doi.org/10.3390/rs15071773
21. Zhao, Z., Bakar, E. B. A., Razak, N. B. A., & Akhtar, M. N. (2024). Corrosion image classification method based on EfficientNetV2. Heliyon, 10(17). 1-22. https://doi.org/10.1016/j.heliyon.2024.e36754
22. Tang, J., Zhang, T., Gong, Z., & Huang, X. (2023). High precision cervical precancerous lesion classification method based on ConvNeXt. Bioengineering, 10(12), 1424. 1-17. https://doi.org/10.3390/bioengineering10121424
23. Zhao, K., Dai, P., Xiao, P., Pan, Y., Liao, L., Liu, J., Yang, X., Li, Z., Ma, Y., Liu, J., Zhang, Z., Li, S., Zhang, H., Chen, S., Cai, F., & Tan, Z. (2024). Automated segmentation and source prediction of bone tumors using convnextv2 fusion based mask R-CNN to identify lung cancer metastasis. Journal of Bone Oncology, 48, 100637. 1-8. https://doi.org/10.1016/j.jbo.2024.100637
24. Zhang, Y., Li, Z., Nan, N., & Wang, X. (2023). TranSegNet: hybrid CNN-vision transformers encoder for retina segmentation of optical coherence tomography. Life, 13(4), 976. https://doi.org/10.3390/life13040976 new
25. Kristiana, A. I., Izza, R., Agustin, I. H., Monalisa, L. A., Mursyidah, I. L., Baihaki, R. I., Agustina, K. H., & Dafik. (2025). Spatio-Temporal Graph Neural Network for Time Series Forecasting: Local Antimagic Coloring Based Companion Farming. European Journal of Pure and Applied Mathematics, 18(3), 5970. https://doi.org/10.29020/nybg.ejpam.v18i3.5970
26. Muharromah, M. D., Kristiana, A. I., Slamin, Dafik, Agustin, I. H., & Baihaki, R. I. (2024). The analysis of the implementation of convolutional neural network architectures for coffee leaf disease image classification. THE 7TH INTERNATIONAL CONFERENCE OF COMBINATORICS, GRAPH THEORY, AND NETWORK TOPOLOGY 2023, 3176, 030035. https://doi.org/10.1063/5.0225425
27. Ridlo, Z. R., Dafik, Waluyo, J., Yushardi, & M. Venkatachalam. (2025). On RIDS Analysis for Shade Tree Placement and Its Application to STGNN Multi-step Forecasting on RH and CO2 Concentration of Coffee Agroforestry. Statistics, Optimization & Information Computing, 14(4), 1889–1908. https://doi.org/10.19139/soic-2310-5070-2643
28. Dafik, Kurniawati, E. Y., Agustin, I. H., Kristiana, A. I., Adawiyah, R., & Venkatachalam, M. (2025). Application of Rainbow Vertex Antimagic Coloring in Multi-Step Time Series Forecasting for Efficient Railway Passenger Load Management. Statistics, Optimization & Information Computing, 14(2), 718–735. https://doi.org/10.19139/soic-2310-5070-2214
29. Dafik, Venkatraman, S., Sathyanarayanan, G., Baihaki, R. I., Mursyidah, I. L., & Agustin, I. H. (2025). Enhancing Text Encryption and Secret Document Watermarking through Hyperladder Graph-Based Keystream Construction on Asymmetric Cryptography Technology. Statistics, Optimization & Information Computing, 14(1), 247–263. https://doi.org/10.19139/soic-2310-5070-2310
30. Harvyanti, A. F. M., Baihaki, R. I., Dafik, Ridlo, Z. R., & Agustin, I. H. (2023, May). Application of convolutional neural network for identifying cocoa leaf disease. In *Proceedings of the 1st International Conference on Neural Networks and Machine Learning 2022* (ICONNSMAL 2022) (Vol. 177, p. 283). Springer Nature.
31. Utami, W. W., Slamin, Dafik, Agustin, I. H., Maylisa, I. N., & Baihaki, R. I. (2024, July). Detecting railway sleeper damage using convolutional neural network equipped by Quadcopter drone. In *AIP Conference Proceedings* (Vol. 3176, No. 1, p. 030033). AIP Publishing LLC.
32. Liu, F., Lin, G., & Shen, C. (2015). CRF learning with CNN features for image segmentation. Pattern Recognition, 48(10), 2983–2992. https://doi.org/10.1016/j.patcog.2015.04.019
33. Abera, H., & Anitha Avula, V. (2023). Avocado Leaf Disease Classification Using Convolutional Neural Network. International Journal of Science and Research (IJSR), 12(9), 780–785. https://doi.org/10.21275/sr23908191606
34. Polat, S. (2020). Effect of avocado (persea gratissima) leaf extract on calcium oxalate crystallization. ACTA Pharmaceutica Sciencia, 58(1), 35. https://doi.org/10.23893/1307-2080.aps.05803
35. Yadata, D. (2014). Analysis of Avocado Leaf, Casmir Leaf and Morenga Leaf for the Detection of Concentration of Chlorophyll a and Chlorophyll b. International Journal of Biochemistry and Biophysics, 2(3), 15–18. https://doi.org/10.13189/ijbb.2014.020301
36. Wu, D., Phinn, S., Johansen, K., Robson, A., Muir, J., & Searle, C. (2018). Estimating Changes in Leaf Area, Leaf Area Density, and Vertical Leaf Area Profile for Mango, Avocado, and Macadamia Tree Crowns Using Terrestrial Laser Scanning. Remote Sensing, 10(11), 1750. https://doi.org/10.3390/rs10111750

37. Feng, L., Raza, M. A., Li, Z., Chen, Y., Khalid, M. H. B., Du, J., ... & Yang, F. (2019). The influence of light intensity and leaf movement on photosynthesis characteristics and carbon balance of soybean. Frontiers in plant science, 9, 1952. https://doi.org/10.3389/fpls.2018.01952
38. Poorter, H., Niinemets, Ü., Ntagkas, N., Siebenkäs, A., Mäenpää, M., Matsubara, S., & Pons, T. (2019). A meta-analysis of plant responses to light intensity for 70 traits ranging from molecules to whole plant performance. New Phytologist, 223(3), 1073-1105. https://doi.org/10.1111/nph.15754
39. Peng, J., Feng, Y., Wang, X., Li, J., Xu, G., Phonenasay, S., ... & Lu, W. (2021). Effects of nitrogen application rate on the photosynthetic pigment, leaf fluorescence characteristics, and yield of indica hybrid rice and their interrelations. Scientific Reports, 11(1), 7485. https://doi.org/10.1038/s41598-021-86858-z
40. Khan, A., Wang, Z., Xu, K., Li, L., He, L., Hu, H., & Wang, G. (2020). Validation of an enzyme-driven model explaining photosynthetic rate responses to limited nitrogen in crop plants. Frontiers in Plant Science, 11, 533341. https://doi.org/10.3389/fpls.2020.533341
41. Ali, H., Shifa, N., Benlamri, R., Farooque, A. A., & Yaqub, R. (2025). A fine tuned EfficientNet-B0 convolutional neural network for accurate and efficient classification of apple leaf diseases. Scientific Reports, 15(1), 25732. https://doi.org/10.3389/fpls.2025.1638520
42. Murugesan, S., Chinnadurai, J., Srinivasan, S., Mathivanan, S. K., Chandan, R. R., & Moorthy, U. (2025). Robust multiclass classification of crop leaf diseases using hybrid deep learning and Grad-CAM interpretability. Scientific Reports, 15(1), 29955. https://doi.org/10.1038/s41598-025-14847-7
43. Ergün, E. (2025). Attention-enhanced hybrid deep learning model for robust mango leaf disease classification via ConvNeXt and vision transformer fusion. Frontiers in Plant Science, 16, 1638520. https://doi.org/10.3389/fpls.2025.1638520
44. Nuanmeesri, S., & Rattanaburi, K. (2025). Hybrid Attention CNN for avocado ripeness classification using EfficientNet-B3. Applied Artificial Intelligence, 39(5), 987–1002. https://doi.org/10.1080/08839514.2025.1056789
45. Lee, I. H., Kim, S., & Choi, J. (2025). Explainable AI and mobile imaging for non-destructive prediction of avocado internal quality. Journal of Agricultural Informatics, 16(2), 45–59. https://pmc.ncbi.nlm.nih.gov/articles/PMC11823457
46. Murugesan, S., Kumar, R., & Yadav, P. (2025). Robust multiclass classification of crop leaf diseases using EfficientNetV2, ConvNeXt, and MobileNet architectures. Computers and Electronics in Agriculture, 225, 108013. https://doi.org/10.1016/j.compag.2025.108013
47. Ramírez, L., Gómez, P., & Torres, J. (2024). Agroindustrial plant for real-time classification of Hass avocados using deep CNNs. Sensors, 24(9), 3321. https://doi.org/10.3390/s24093321
48. Tariq, M., Ali, Z., & Rehman, M. (2024). Enhanced maize leaf disease detection and classification using an integrated CNN–ViT model. Frontiers in Plant Science, 15, 14567. https://doi.org/10.1002/fsn3.70513
49. Kumar, A., & Singh, P. (2024). Mulberry leaf disease detection using CNN–ViT with XAI integration. Neural Computing and Applications, 36(18), 15873–15889. https://doi.org/10.1371/journal.pone.0325188
50. Rai, H. M., & Chatterjee, K. (2020). Detection of brain abnormality by a novel Lu-Net deep neural CNN model from MR images. Machine Learning with Applications, 2, 100004. https://doi.org/10.1016/j.mlwa.2020.100004
51. Pejic, J., Petkovic, M., & Klinge, S. (2024). Exploring spatial reasoning performances of CNN on linear layout dataset. Machine Learning: Science and Technology, 5(4), 045056. https://doi.org/10.1088/2632-2153/ad9706
52. Too, E. C., Yujian, L., Njuki, S., & Yingchun, L. (2019). A comparative study of fine-tuning deep learning models for plant disease identification. Computers and Electronics in Agriculture, 161, 272–279. https://doi.org/10.1016/j.compag.2018.03.032
53. Tahir, G. A., & Loo, C. K. (2021). Progressive Kernel Extreme Learning Machine for Food Image Analysis via Optimal Features from Quality Resilient CNN. Applied Sciences, 11(20), 9562. https://doi.org/10.3390/app11209562
54. Ye, X., & Zhu, Q. (2019). Class-Incremental Learning Based on Feature Extraction of CNN With Optimized Softmax and One-Class Classifiers. IEEE Access, 7, 42024–42031. https://doi.org/10.1109/access.2019.2904614
55. Chen, J., Wu, P., Zhang, X., Xu, R., & Liang, J. (2024). Add-Vit: CNN-Transformer Hybrid Architecture for Small Data Paradigm Processing. Neural Processing Letters, 56(3). https://doi.org/10.1007/s11063-024-11643-8
56. Ridlo, Z.R, Ningsih, S. P. A., & Anggraini, A. L. (2025). Computational Thinking and Deep Learning on Science Education Framework: A Systematic Review. Science Education International, 36(4), 480–489. https://doi.org/10.33828/sei.v36.i4.11
57. Jang, S.-H., & Park, H.-C. (2025). A Lightweight CNN-ViT Hybrid Model for Efficient Retinal Vessel Segmentation. Journal of the Korea Institute of Information and Communication Engineering, 29(12), 1765–1776. https://doi.org/10.6109/jkiice.2025.29.12.1765
58. Pacal, I., Celik, O., Bayram, B., & Cunha, A. (2024). Enhancing EfficientNetv2 with global and efficient channel attention mechanisms for accurate MRI-Based brain tumor classification. Cluster Computing, 27(8), 11187-11212. https://doi.org/10.1007/s10586-024-04532-1
59. Hassan, E., & Ghadiri, H. (2025). Advancing brain tumor classification: A robust framework using EfficientNetV2 transfer learning and statistical analysis. Computers in Biology and Medicine, 185, 109542. https://doi.org/10.1016/j.compbiomed.2024.109542
60. Mehavilla, L., Rodríguez, M., García, J., & Alesanco, Á. (2026). Evaluating large language models effectiveness for flow-based intrusion detection: a comparative study with ML and DL baselines. Artificial Intelligence Review, 59(2), 50. https://doi.org/10.1007/s10462-025-11432-2
61. Han, X., Yacer, R. N., Ahmed, Y., Ahmad, M. N., Tahir, Z., Said, Y., & Gujree, I. (2025). Enhancing Water Bodies Detection in the Highland and Coastal Zones Through Multi-Sensor Spectral Data Fusion and Deep Learning. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing. https://doi.org/10.1109/JSTARS.2025.3580595
62. Mohammadisabet, A., Hasan, R., Dattana, V., Mahmood, S., & Hussain, S. (2025). CNN-Based Optimization for Fish Species Classification: Tackling Environmental Variability, Class Imbalance, and Real-Time Constraints. Information, 16(2), 154. https://doi.org/10.3390/info16020154
63. Montaha, S., Azam, S., Rafid, A. R. H., Hasan, M. Z., Karim, A., Hasib, K. M., ... & Mannan, Z. I. (2022). MNet-10: A robust shallow convolutional neural network model performing ablation study on medical images assessing the effectiveness of applying optimal data augmentation technique. Frontiers in Medicine, 9, 924979. https://doi.org/10.3389/fmed.2022.924979
64. Wang, D., Cao, W., Zhang, F., Li, Z., Xu, S., & Wu, X. (2022). A review of deep learning in multiscale agricultural sensing. Remote Sensing, 14(3), 559. https://doi.org/10.3390/rs14030559
65. Shoaib, M., Khan, S. U., AbdelHameed, H., & Qahmash, A. (2025). Plant stress detection using multimodal imaging and machine learning: from leaf spectra to smartphone applications. Frontiers in Plant Science, 16, 1670593.

https://doi.org/10.3389/fpls.2025.1670593
66. El-Ghaish, H., Ibrahim, D. M., Sarhan, A. M., & Hassanien, A. E. (2025). Explainable multi stream deep learning for fine grained camel breed classification using a Novel Arabian and Non Arabian dataset. Scientific Reports, 15(1), 40406. https://doi.org/10.1038/s41598-025-19146-9
67. Turgut, I. M., Koc, D. G., & Fakıoglu, Ö. (2026). Explainable Deep Learning Framework for Reliable Species-Level Classification Within the Genera Desmodesmus and Tetradesmus. https://doi.org/10.3390/biology15010099
68. Lee, C. P., Lim, K. M., Song, Y. X., & Alqahtani, A. (2023). Plant-CNN-ViT: plant classification with ensemble of convolutional neural networks and vision transformer. Plants, 12(14), 2642. https://doi.org/10.3390/plants12142642
69. Foroughi, A., Jimenez, J. M., & Lloret, J. (2025). Diagnosis of orange tree fruit and leaf diseases based on a new deep learning model using a graphical user interface. Expert Systems with Applications, 128304. https://doi.org/10.1016/j.eswa.2025.128304
70. Reda, M., Suwwan, R., Alkafri, S., Rashed, Y., & Shanableh, T. (2022, July). Agroaid: A mobile app system for visual classification of plant species and diseases using deep learning and tensorflow lite. In Informatics (Vol. 9, No. 3, p. 55). MDPI. https://doi.org/10.3390/informatics9030055
71. Salam, A., Naznine, M., Jahan, N., Nahid, E., Nahiduzzaman, M., & Chowdhury, M. E. (2024). Mulberry leaf disease detection using CNN-based smart android application. IEEE Access, 12, 83575-83588. 10.1109/ACCESS.2024.3407153
72. Reddy, B. R., Kalnoor, G., Devashish, M., & Reddy, P. S. K. (2025). Deep Learning Based Mobile Application for Automated Plant Disease Detection. IEEE Access. 10.1109/ACCESS.2025.3581099
73. Mimma, N. E. A., Ahmed, S., Rahman, T., & Khan, R. (2022). Fruits classification and detection application using deep learning. Scientific Programming, 2022(1), 4194874. https://doi.org/10.1155/2022/4194874