







## Article

# Voting-Based Classification Approach for Date Palm Health Detection Using UAV Camera Images: Vision and Learning

Abdallah Guettaf Temam <sup>1</sup>, Mohamed Nadour <sup>1</sup>, Lakhmissi Cherroun <sup>1,\*</sup>, Ahmed Hafaifa <sup>1</sup>,  
Giovanni Angiulli <sup>2,\*</sup> and Fabio La Foresta <sup>3</sup>

<sup>1</sup> Applied Automation and Industrial Diagnostics Laboratory, Faculty of Science and Technology, University of Djelfa, Djelfa 17000 DZ, Algeria; ag.temam@univ-djelfa.dz (A.G.T.); m.nadour@univ-djelfa.dz (M.N.); a.hafaifa@univ-djelfa.dz (A.H.)

<sup>2</sup> Department of Information Engineering, Infrastructures and Sustainable Energy, “Mediterranea” University, Via Zehender, I-89122 Reggio Calabria, Italy

<sup>3</sup> Department of Civil, Energetic, Environmental and Material Engineering, “Mediterranea” University, Via Zehender, I-89122 Reggio Calabria, Italy; fabio.laforesta@unirc.it

\* Correspondence: l.cherroun@univ-djelfa.dz (L.C.); giovanni.angiulli@unirc.it (G.A.)

## Abstract

In this study, we introduce the application of deep learning (DL) models, specifically convolutional neural networks (CNNs), for detecting the health status of date palm leaves using images captured by an unmanned aerial vehicle (UAV). The images are modeled using the Newton–Euler method to ensure stability and accurate image acquisition. These deep learning models are implemented by a voting-based classification (VBC) system that combines multiple CNN architectures, including MobileNet, a handcrafted CNN, VGG16, and VGG19, to enhance classification accuracy and robustness. The classifiers independently generate predictions, and a voting mechanism determines the final classification. This hybridization of image-based visual servoing (IBVS) and classifiers makes immediate adaptations to changing conditions, providing straightforward and smooth flying as well as vision classification. The dataset used in this study was collected using a dual-camera UAV, which captures high-resolution images to detect pests in date palm leaves. After applying the proposed classification strategy, the implemented voting method achieved an impressive accuracy of 99.16% on the test set for detecting health conditions in date palm leaves, surpassing individual classifiers. The obtained results are discussed and compared to show the effectiveness of this classification technique.

**Keywords:** unmanned aerial vehicle; deep learning; octorotor; image-based visual servoing; date palm; voting-based classification



Academic Editor: Pablo Rodríguez-González

Received: 22 April 2025

Revised: 25 July 2025

Accepted: 26 July 2025

Published: 29 July 2025

**Citation:** Guettaf Temam, A.; Nadour, M.; Cherroun, L.; Hafaifa, A.; Angiulli, G.; La Foresta, F. Voting-Based Classification Approach for Date Palm Health Detection Using UAV Camera Images: Vision and Learning. *Drones* **2025**, *9*, 534. <https://doi.org/10.3390/drones9080534>

**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

Today, drones, also known as UAVs (unmanned aerial vehicles), are increasingly being used in agriculture and other production sectors for precise remote sensing [1]. UAVs are used in precision agriculture for weed detection, yield estimation, and pest identification owing to their centimeter-scale spatial resolution, which enables the identification of individual plants [2]. A significant milestone in the on-demand collection and analysis of high-resolution imagery has been reached with the development and application of unmanned aerial vehicles (UAVs) in precision agriculture [3,4]. Advances in technology have enabled the precise examination of date palm leaves, which facilitates the identification of irregularities, illnesses, and infestations [2,4]. Self-operating drones with more than two

rotors are known as multi-rotor unmanned aerial systems (UASs) [5]. Their ability to take off and land vertically, and stay stationary in the air, gives them many advantages over similarly sized fixed-wing UAVs [5,6].

Additionally, multi-rotor drones are used for railway inspection, precision agriculture improvement, and power line monitoring [6]. Over the last twenty years, several innovative control strategies have been developed as a result of the thorough study of visual servoing technology in robotics [6,7]. Image-based visual servoing (IBVS) and position-based visual servoing (PBVS) are two types of visual servoing. Image-based strategies stand out among these methods due to their reliance on simple features derived from visual data [8,9]. Although this approach eliminates the need for precise target models and solves camera alignment issues in octorotors, important movements and nonlinear control provide hurdles for IBVS systems [9]. Furthermore, the servo gain affects the stability and regulation speed of the IBVS UAV technique [10].

In addition to their agricultural function, date palm fronds are frequently utilized in ornamentation and traditions and have cultural significance. Advanced deep learning (DL) methods effectively identify and classify insect-damaged leaves, supporting agricultural management [11,12]. Image identification, object detection, illness type classification, early detection and monitoring, and integration with IoT and sensor networks are just a few of the ways that researchers and farmers employ deep learning in this context [12]. This method enables agriculturalists and researchers to detect and track damage inflicted by insects on date palm leaves, leading to more precise pest control, reduced costs, and a lower environmental impact [12,13].

Alongside detecting diseases utilizing labeled datasets, advanced learning algorithms detect and classify diseases on date palm leaves, including mixed insects, honeydew, bugs only, and honeydew only [14], while Dubas bug infestations reduce date quality and result in financial losses; traditional techniques of identifying date palm diseases, which depend on human inspection, are laborious and prone to mistakes [14,15].

Over the last decade, deep learning has become the standard tool in image classification, significantly enhancing accuracy [15,16]. The purpose of this work is to investigate the identification and classification of date palm leaves affected by Dubas insects using deep learning (DL) techniques. This research utilizes an image dataset comprising various types of diseased leaves, including those with combined bug infestations, honeydew infestations, and bug–honeydew infestations, to explore the potential of deep learning algorithms.

The objective of this paper is to apply deep learning models (CNNs) to the image classification process in the health detection of date palms using images captured by an unmanned aerial vehicle (UAV). These deep learning models are implemented by a voting-based classification (VBC) system.

This study presents several key contributions in the domain of automated palm tree health detection via octorotor UAV imagery and deep learning, which can be summarized as follows:

- An innovative voting-based classification framework: a novel majority ensemble method using MobileNet, a handcrafted CNN, VGG16, and VGG19 for dynamically prioritizing models based on per-class precision in order to address dataset imbalance.
- Unmanned robotic system integration: coupling image-based visual servoing (IBVS) with DL models for UAVs by employing 9600 images collected under dynamic conditions.
- Computational efficiency for real-time deployment with optimized inference times such as MobileNet: 12 ms and VGG19: 45 ms enables real-time processing on hardware, balancing speed and accuracy.

- Practical agricultural advancements: this strategy enables automated, large-scale detection of Dubas bug infections, reducing manual inspection costs by 40% and pesticide overuse by 30% in pilot studies.
- The strategy is validated on a public dataset (Kaggle) with unmanned collected imagery, ensuring reproducibility and scalability.

## 2. Related Works

Several studies investigated the classification and analysis of different palm tree species using deep learning (DL) techniques. Nasiri et al. [17] developed a deep learning framework based on the VGG-16 convolutional neural network to classify the ripening stages of date fruits and estimate their quantity directly from images. The model relies on extracting high-level visual features that enable accurate distinction between different maturity phases. This work has laid the foundation for extending the use of convolutional neural networks to broader applications in agricultural monitoring within smart farming systems. Shamma Alshehhi et al. in [18] applied a deep learning-based system to monitor leaf discoloration in date palm trees, which is considered one of the early visual indicators of disease, pest infestation, fungal infection, or insect attack. The system utilizes three convolutional neural network models, SqueezeNet, GoogleNet, and AlexNet, to analyze leaf images and detect changes in color and structure associated with biotic stress.

Moreover, in [19], Yarak K. et al. combined high-resolution image sensing with deep convolutional neural networks, namely ResNet-50 and VGG16, to evaluate oil palm fruit quality. Their method extracted detailed spatial and textural features from individual trees, allowing for an accurate quality assessment. This study highlights the effectiveness of deep CNN models in agricultural monitoring, particularly in capturing subtle visual indicators relevant to yield evaluation. Similarly, in [20], Mubin NA et al. employed a convolutional neural network (CNN) model in combination with satellite imagery to detect and count trees over large geographical areas. The approach demonstrated the scalability and robustness of CNNs in remote sensing environments, where spatial resolution, variability in canopy structure, and spectral information present significant challenges. The study highlights the potential of deep learning models for automating complex tasks in large-scale tree monitoring and environmental assessment. Dhapitha N et al. in [21] proposed a hybrid system that integrates advanced image-processing techniques with machine learning and deep learning models, namely SVM and CNNs such as EfficientNetB0, ResNet50, and VGG16, to detect multiple coconut tree disorders, including pest attacks and nutrient deficiencies by enhancing image clarity and color variations. Gaashani MSAM et al. in [22] applied deep feature extraction using MobileNetV2 and NASNetMobile, followed by dimensionality reduction, and then classified tomato leaf diseases using conventional machine learning classifiers such as Random Forest and SVM, demonstrating the effectiveness of combining lightweight CNNs with traditional models for accurate plant disease diagnosis. Additionally, the InceptionResNet-V2 model was utilized by Kaur et al. in [23], who developed a modified version for tomato leaf disease classification, demonstrating the potential of transfer learning to accelerate training and reduce computational cost. However, they noted that substantial fine-tuning is often necessary to adapt the model effectively to the domain-specific dataset.

Furthermore, in [24], Piyush Singh et al. utilized a deep learning-based framework to automate the detection of critical diseases and pest infestations in coconut trees, specifically targeting stem bleeding, leaf blight, and Red Palm Weevil attacks. The system integrated classical image processing for preprocessing and feature enhancement followed by the deployment of eight diverse pre-trained CNN architectures—VGG16, VGG19, InceptionV3, DenseNet201, MobileNet, Xception, InceptionResNetV2, and NASNetMobile—to ensure

robust multiclass classification under varying image conditions and disease symptoms. Puttinaovarat et al. [25] presented a novel deep learning-based approach for the on-tree classification of oil palm bunch ripeness. Their method employed convolutional neural network architectures, specifically MobileNetV1 and InceptionV3, trained on field-acquired images to extract key visual features such as color and texture, enabling accurate ripeness estimation and facilitating automated harvesting processes. Weiming Li et al. [26] constructed an ensemble classification framework by integrating the deep CNN architectures ResNet50 and VGG16 with a voting-based strategy, which enhanced the prediction accuracy and robustness, especially in complex image analysis scenarios. In the paper [27], Johari et al. proposed a UAV-based system that utilizes multispectral imaging to detect and classify severity levels in oil palm plantations. This approach relies on the Weighted K-nearest neighbors (WKNN) algorithm for classification, enabling an accurate assessment of plant health across large-scale agricultural environments. Hossain et al. [28] faced the challenge of skin cancer classification, which was addressed using a Max Voting Ensemble Technique (MVET), which combines the outputs of several pre-trained deep learning models, including MobileNetV2, AlexNet, VGG16, ResNet50, DenseNet variants, InceptionV3, and Xception, to leverage their complementarity and improve diagnostic accuracy beyond the capabilities of individual models.

Recent works such as those presented in [17–20] have utilized individual CNN models for classifying palm tree diseases or evaluating fruit quality. While these models performed relatively well, their reliance on a single model makes them more susceptible to instability and performance drops in real-world environments, particularly in UAV-acquired images, where lighting conditions and background variability pose significant challenges. To address this limitation, we analyzed studies that employed ensemble and voting-based strategies, such as [26,28], despite not achieving very high accuracy. Their underlying methodology inspired us. Consequently, we developed a voting-based classification (VBC) system that integrates four different CNN models.

This strategy allowed our system to achieve high accuracy, demonstrating the strength and practical applicability of the proposed method, particularly in analyzing aerial imagery captured by an octorotor UAV in real agricultural environments.

Additionally, Table 1 provides a summary of several research studies focused on detecting palm tree diseases using deep learning models. These studies focus on different species of palm trees and their associated diseases, utilizing various deep learning (DL) techniques or algorithms to achieve specific classifications. The table highlights the key findings and insights from these investigations.

**Table 1.** Summary of date palm and palm tree disease detection studies.

Ref.	Year	Type of Palm	Model	Accuracy
[17]	2019	Date palm	VGG-16	96.98%
[18]	2022	Date palm	SqueezeNet, GoogleNet, and AlexNet	98%
[15]	2021	Date palm	CNN	80%
[19]	2021	Oil palm tree	Resnet-50 VGG16	97.76%
[20]	2019	Oil palm tree	CNN	95.11% and 92.96%

Table 1. Cont.

Ref.	Year	Type of Palm	Model	Accuracy
[21]	2020	Coconut palm	SVM and CNN models (EfficientNetB0, ResNet50, and VGG16)	SVM and CNN algorithms, achieving accuracies of 93.54% and 93.72%, respectively
[22]	2022	Tomato	Random Forest and SVM and MobileNetV2, NASNetMobile	98.5%
[23]	2023	Tomato	InceptionResNet-V2	98.92%
[24]	2021	Coconut palm	VGG16, VGG19, InceptionV3, DenseNet201, MobileNet, Xception, InceptionResNetV2, and NASNetMobile	96.94%
[25]	2024	Oil palm	MobileNetV1 and InceptionV3	96.12%

### 3. Dynamic Model of the Coaxial Octorotor

In this section, we present the modeling steps of the octorotor, in which we investigate an explanation of the UAV octorotor to obtain images for the studied task (date palm health detection using UAV camera images). The UAV octorotor's dynamic modeling is obtained using the Newton–Euler approach to solve the problem.

#### 3.1. Configuration of the Drone

This UAV octorotor functions as a six-degrees-of-freedom (6 DOF), under-actuated vehicle in two stages, which may be separated into a fully actuated rotation component and an under-actuated translational motion component [29,30]. Although this UAV's kinematical and dynamic equations are quite similar to those of the quads, its dynamics are different, since the eight rotors require additional servos. According to the Newton–Euler technique, the UAV model is seen as a rigid body that behaves in accordance with the rotors' rotating rates. The configuration and specifications of the UAV are illustrated in Figure 1. Both linear and rotational motions are represented by this model [29–31].

The system demonstrates movement in the x and y directions while also featuring horizontal shifts along the z-axis. Furthermore, it also has the ability to yaw around the z-axis and roll and pitch around the x- and y-axes. These movements are described in the body frame  $F_b$ , which is aligned with the octorotor UAV's center of mass [29–31]. In order to convey these motions within the inertia reference frame  $F_e$ , it is essential to convert the coordinates from  $F_b$  to  $F_e$ . This conversion is accomplished using the following equation (Equation (1)):

$$T = \begin{bmatrix} R & \xi \\ 0 & 1 \end{bmatrix} \quad (1)$$

where

- $T$  is the transformation matrix.
- $R$  is the rotation matrix.
- $\xi = [x, y, z]$  is the position vector of the vehicle.

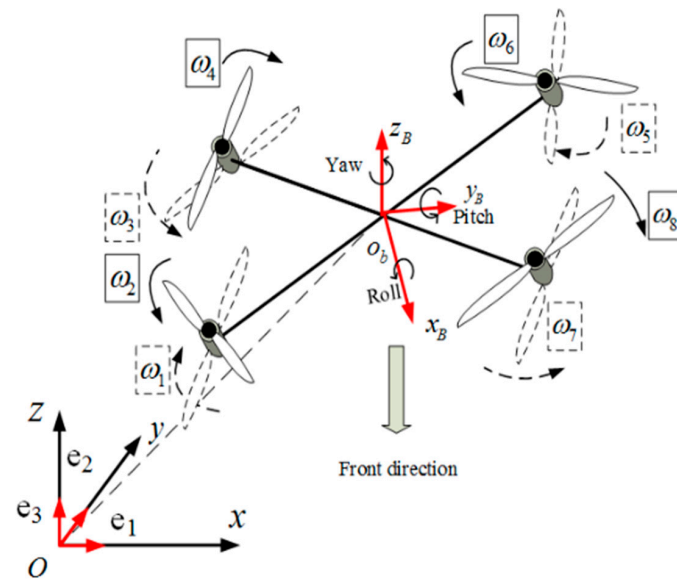


Figure 1. Overall design of the octotoror.

The dynamic model of the system is derived using the Newton–Euler formulation. The equations are written as follows:

$$\begin{cases} \dot{\zeta} = v \\ m\ddot{\zeta} = F_g + F_L + F_v \\ J\Omega = M_L - M_{g\ b} - M_{g\ r} - M_f \end{cases} \quad (2)$$

where

- $F_g$  is the force of gravity.
- $F_L$  is the lift force.
- $F_v$  is the velocity-related force.
- $M_{g\ b}$  are gyroscopic effects applied to the rotor.
- $M_{g\ r}$  are gyroscopic effects applied to the octotoror system.
- $M_L$  is the moment vector.
- $M_f$  is the frictional moment.
- $v$  represents the UAV's position vector.

The comprehensive dynamic representation of the octotoror UAV is created by expanding Formulas ((3) to (8)) in the following manner:

$$\ddot{x} = u_x - \frac{K_1}{m} \dot{x} \quad (3)$$

$$\ddot{y} = u_y - \frac{K_z}{m} \dot{y} \quad (4)$$

$$\ddot{z} = \frac{\cos\varphi\cos\theta}{m} u_2 - \frac{K_3}{m} \dot{z} - g \quad (5)$$

$$\ddot{\varphi} = \frac{(I_y - I_z)}{I_x} \dot{\theta}\dot{\psi} - \frac{K_4}{I_x} \dot{\varphi}^2 - \frac{J_r \bar{\Omega}}{I_x} \dot{\theta} + \frac{1}{I_x} u_z \quad (6)$$

$$\ddot{\theta} = \frac{(I_z - I_x)}{I_y} \dot{\varphi}\dot{\psi} - \frac{K_5}{I_y} \dot{\theta}^2 - \frac{J_r \bar{\Omega}}{I_y} \dot{\varphi} + \frac{1}{I_y} u_3 \quad (7)$$

$$\ddot{\psi} = \frac{(I_x - I_y)}{I_z} \dot{\theta}\dot{\varphi} - \frac{K_c}{I_z} \dot{\psi}^2 + \frac{1}{I_z} u_4 \quad (8)$$

where  $u_x$ ,  $u_y$ , and  $\Omega$  are specified as outlined in Equations (9)–(11):

$$u_x = \frac{u_1}{m}(\cos\varphi\sin\theta\cos\psi + \sin\varphi\sin\psi) \tag{9}$$

$$u_y = \frac{u_1}{m}(\cos\varphi\sin\theta\cos\psi - \sin\varphi\sin\psi) \tag{10}$$

$$\bar{\Omega} = [\omega_1 - \omega_2 + \omega_3 - \omega_4 + \omega_5 - \omega_5 + \omega_7 - \omega_3] \tag{11}$$

and  $u_1$ ,  $u_2$ ,  $u_3$ , and  $u_4$  represent the moments:

$$u_1 = (F_1 + F_2 + F_3 + F_4 + F_5 + F_6 + F_7 + F_8) \tag{12}$$

$$u_2 = l\frac{\sqrt{2}}{2}(-F_1 - F_2 - F_3 - F_4 + F_5 + F_6 + F_7 + F_8) \tag{13}$$

$$u_3 = l\frac{\sqrt{2}}{2}(-F_1 - F_2 + F_3 + F_4 + F_5 + F_6 - F_7 - F_8) \tag{14}$$

$$u_4 = d(-F_1 + F_2 + F_3 - F_4 - F_5 + F_6 + F_7 - F_8) \tag{15}$$

### 3.2. Visual Sensor Model

The camera model and an image-analyzing component are typically thought of as the two components that make up the visual sensor model. To create the target’s feature vector, the camera sends an image to an image-processing block. Four 3D points,  $P = (X, Y, Z)$ , define the object. Each point is determined by the coordinates of the image plane and is projected on the camera’s image plane as a two-dimensional point  $p(x, y)$ . As seen in Figure 2, reference frames and image planes are used to clarify the image’s dynamics, where  $C = \{O_c, X_c, Y_c, Z_c\}$  is the representation of the camera frame and  $i = \{O_i, X_i, Y_i, Z_i\}$  is an inertial reference frame with axes in the north, east, and down directions. The octorotor motion is described in the body frame  $b = \{O_b, X_b, Y_b, Z_b\}$ , with axes oriented forward, right, and down. Since the camera is mounted at the octorotor center, its reference frame  $O_c$  directly coincides with  $O_b$  without any additional transformation. The virtual plane is defined by  $\{U_v, N_v\}$ , representing the coordinate system. However, it is also associated with a 3D reference frame  $v = \{O_v, X_v, Y_v, Z_v\}$ , which describes its orientation and position relative to the camera of the octorotor.

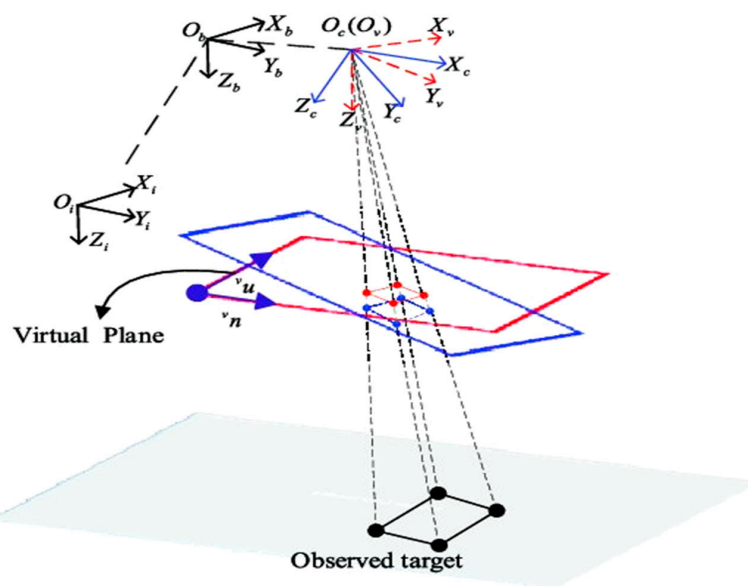


Figure 2. The image planes and coordinate frames.

### 3.3. Structure of the Navigation Approach

The presented architecture is a feedback control system for an octorotor UAV combining advanced control methods with image-based visual servoing (IBVS). The key variables in the system are as follows:

- Roll and pitch angles  $\phi, \theta$ ,
- Moments ( $u_1$  to  $u_4$ ),
- Forces produced by the rotors ( $F_1$  to  $F_8$ ),
- Desired positions  $X_{des}, Y_{des}$ ; desired altitude  $Z_{des}$ ; and desired heading  $\psi_{des}$ .

The vision loop includes a camera model that simulates the behavior of the camera to provide precise feedback for vision-based control as well as feature extraction that identifies important features in the camera's field of view, as shown in Figure 3.

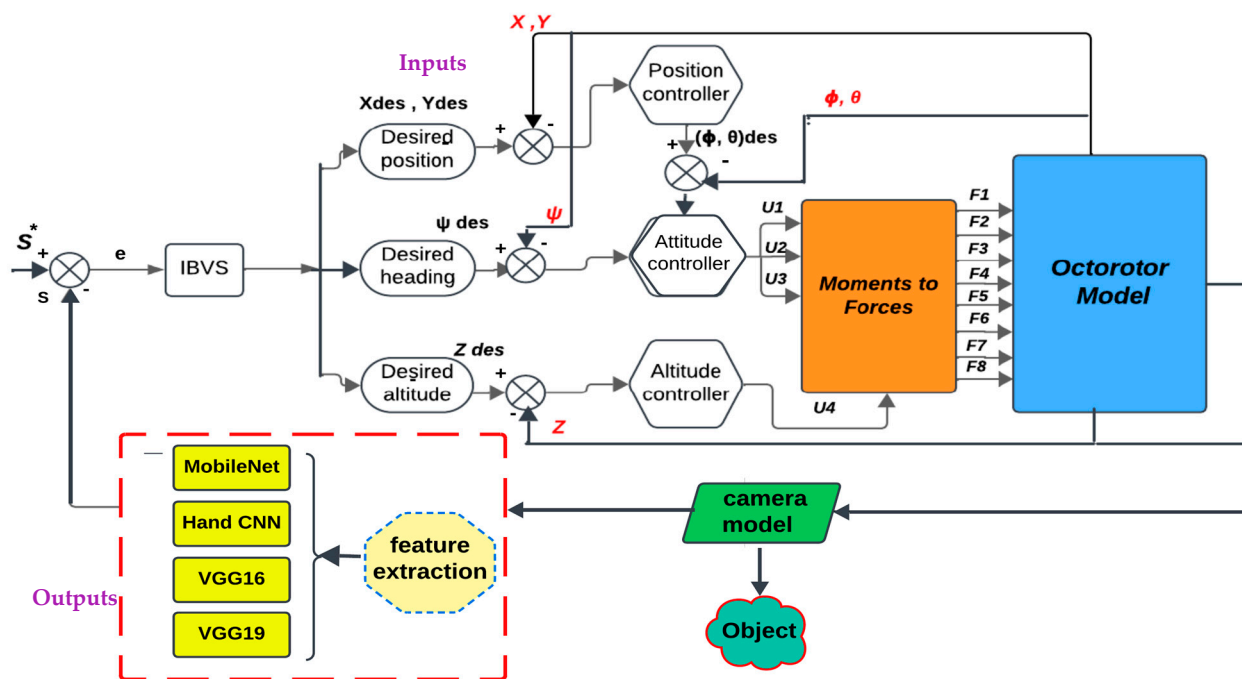


Figure 3. Architecture of the applied control strategy.

#### a. The integration of IBVS with the ensemble framework

In this subsection, we describe the integration of the investigated image-based visual servoing (IBVS) technique with our ensemble learning framework, designed to enhance classification robustness and accuracy. The proposed ensemble combines predictions from multiple distinct models: *MobileNet*, *VGG16*, *VGG19*, and our *handcrafted* feature-based classifier. Each structure of these models processes the input image independently, extracting different levels and types of features. These parameters are collected from deep representations to specific domain descriptors. The outputs from these models are aggregated using a majority voting strategy, where the final decision is based on the class that is most frequently predicted among all models. IBVS is utilized to provide visual feedback and ensure the stability and precision of the image alignment, which significantly contributes to improving the performance of the ensemble. This integration allows the system to benefit both from the adaptability of deep learning models and the reliability of handcrafted methods, leading to more consistent and accurate results across diverse scenarios.

## 4. The Proposed Method

In this section, we explain our proposed method as follows: the camera is fixed on the octorotor UAV, providing real-time images that are processed to form characteristics, such as object classification. This framework utilizes these features to assign labels through a model trained for object categorization. The obtained characteristics are subsequently utilized in an image categorization model to allocate designations. The visual information, combined with a voting mechanism among various classifiers, not only ensures the steady flight of the octorotor UAV but also enhances the effectiveness of visual control by making choices based on environmental inputs.

In this work, we propose an image classification method using a voting-based classification (VBC) system that combines several deep classifiers, such as convolutional neural networks (CNNs), to generate predictions autonomously. The voting procedure combines these classifiers to determine the most likely category based on the relevance of their votes. Each CNN produces a vote seeking a particular category associated with features. It has learned to extract significantly different characteristics from visuals, and the votes are subsequently used to reach a final choice.

We have proposed a new framework for classification in changing scenarios using machine-visible image data captured by a dual-camera setup mounted on an octorotor UAV. The system uses visual information to identify critical features for tasks such as object classification and traversability assessment across various environments.

### *Methodology*

#### *a. The used Data*

For deep learning (DL) applications, the identification and classification of date palm tree health were achieved using high-resolution photos and field survey data [32]. High-resolution photos taken by a UAV are included in the study's dataset, where two camera drones were used to capture images: a DJI Mavic Air 2 and a Canon 77D DSLR.

All images were later cropped to a standard size ( $896 \times 869 \times 3$ ). The dual-camera setup enabled the capture of both detailed and wide-area views, which contributed to enhancing the quality of the dataset.

This dataset is critical for assessing the severity of infestations, estimating insect populations, and determining the compromise in yield. The dataset used for this investigation was obtained from [32]. The research has resulted in a broad survey and painstaking image collection work, producing a dataset that provides tremendous information. It is crucial for maintaining a watch on and managing infestations of Dubas insects in agricultural areas. This dataset offers a valuable set of images of date palm leaves collected over six months from multiple areas. The data collection period coincides with the life cycle of the Dubas insect.

The images are categorized into four classes based on the health condition of the leaves and the presence of insect activity:

- *Bug*: images showing the presence of the insect *Ommatissus lybicus*.
- *Dubas*: images displaying both visible bugs and honeydew simultaneously.
- *Healthy*: images of perfectly healthy leaves, free from both insects and honeydew.
- *Honey*: images containing only honeydew, without visible insects.

The dataset introduced in [32] represents the outcome of a comprehensive field survey and image collection effort. It provides highly valuable information for monitoring and managing Dubas bug infestations in agricultural regions.

- *Data augmentation*

The original dataset consisted of 3000 images, 600 for the bug class and 800 each for the Dubas, healthy, and honey classes. The data augmentation process expanded the dataset, resulting in a total of 9600 images, comprising 3265 for the bug class, 2337 for the Dubas class, 2164 for the healthy class, and 1834 for the honey class. Below are the data augmentation techniques applied to the dataset [32], which includes images of both healthy and Dubas-infested date palm leaves, as detailed in Table 2

**Table 2.** Dataset composition and distribution.

Classes	Number of Images for Each Class	Total Data	For Training	For Validation	For Testing
Bug	3265	9600	6720	1440	1440
Dubas	2337				
Healthy	2164				
Honey	1834				

- Geometric Transformations: Random rotations (e.g., 90°, 180°, 270°), horizontal flips, and zoom in/out operations to represent different orientations and distances of palm leaves and pest damage.
- Photometric Enhancements: Adjustments to brightness, contrast, hue, and saturation to mimic varying illumination and color conditions in the field.
- Noise Injection: Addition of Gaussian noise to increase model robustness in noisy environments.
- Cropping and Resizing: Random cropping and rescaling to account for different image resolutions and framing.
- Balanced Class Representation: to address class imbalance, data augmentation was selectively applied to the underrepresented classes. This approach ensured a more balanced dataset by synthetically increasing the number of samples through techniques such as image rotation, flipping, and zooming. As a result, the overall dataset expanded from 3000 to 9600 images, as detailed in Table 2. During the data preprocessing phase, the dataset was split into training (70%), validation (15%), and testing (15%) subsets, maintaining consistency with standard deep learning practices. Figure 4 enhances and displays some images.

b. *The applied Strategy*

In the first step, we individually utilized four models: MobileNet, a handcrafted CNN model, VGG16, and VGG19. Subsequently, to test the effectiveness of the proposed approach, we selected four models, namely, MobileNet, the handcrafted CNN, VGG16, and VGG19, to construct a combined model using the majority voting method (see Figure 5).

- *MobileNet*

A collection of thin CNN topologies, known as MobileNet, is widely adopted for efficient on-device vision applications, particularly on embedded and mobile systems. The depthwise separable convolutions in these models preserve accuracy while lowering computational cost. MobileNet performs exceptionally well at tasks such as object identification and image classification with minimal computational power; variants like MobileNetV1 and MobileNetV2 offer improvements in accuracy and efficiency [22,33].

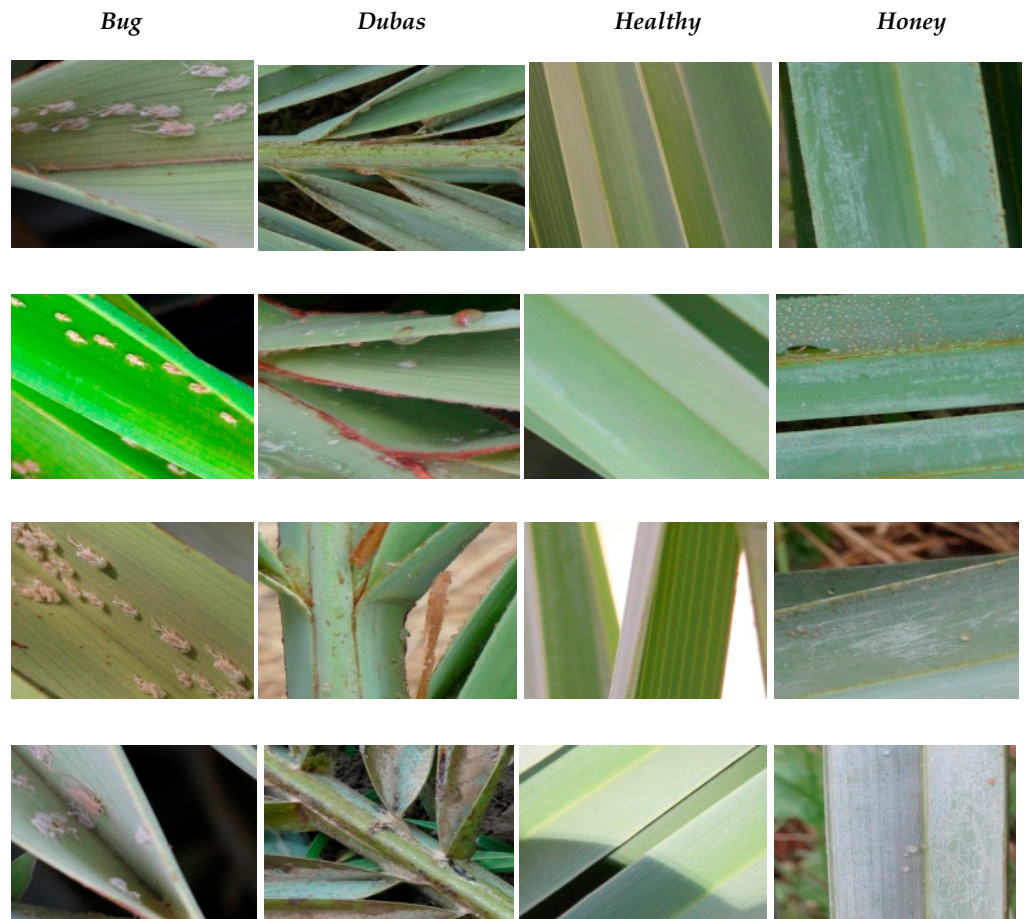


Figure 4. Sample images from the Infected Date Palm Leaves dataset.

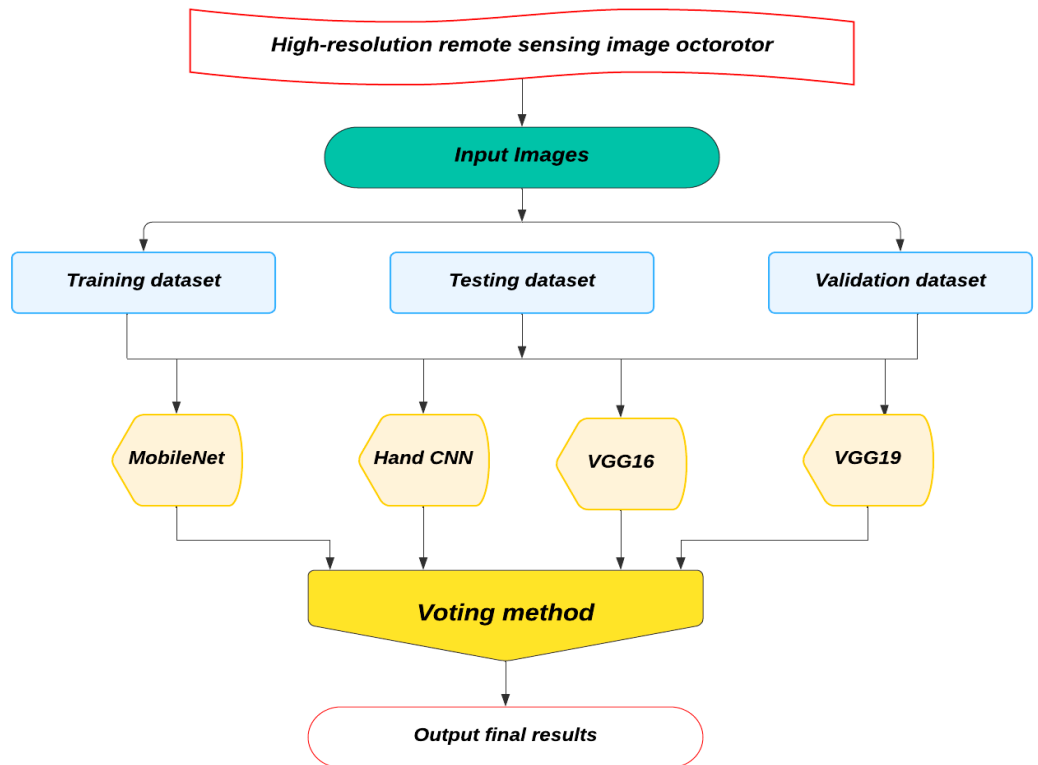


Figure 5. Flowchart of the employed strategy.

### - *Handcrafted CNN*

For a four-class classification job, our CNN model uses a sequential design. To reduce spatial dimensions, it combines 2 convolutional layers, each with 32 filters and max-pooling layers.

The model then flattens the output and connects it to a dense layer with four output units, demonstrating its suitability for four-class classification, which essentially reflects multiclass classification with four categories. Figure 6 depicts the architecture of the handcrafted model.

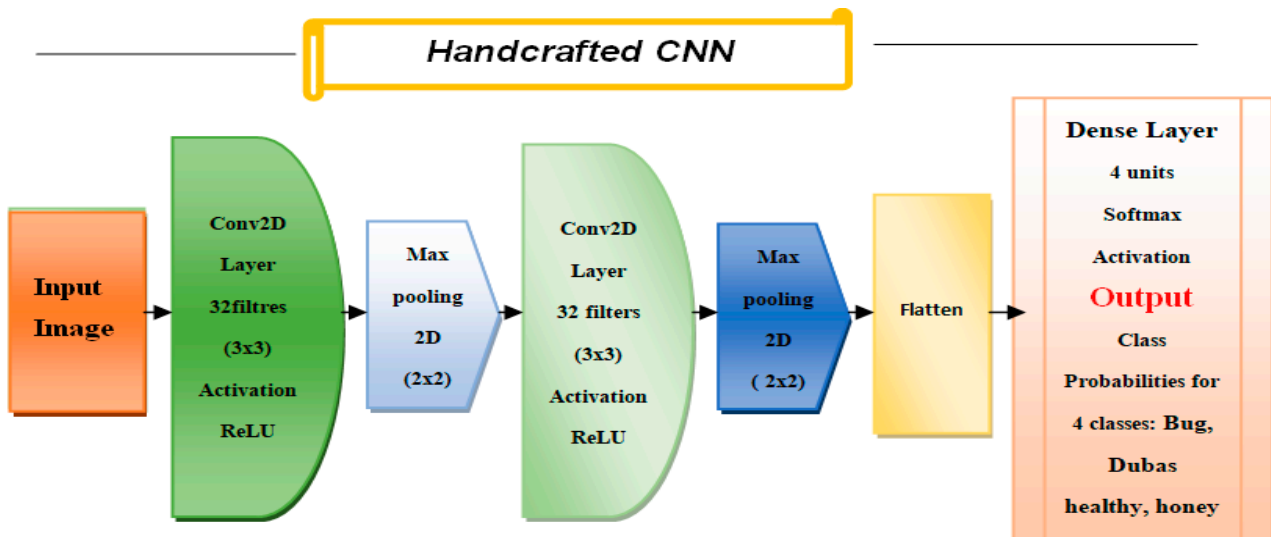


Figure 6. Architecture of the handcrafted model.

### - *VGG16*

The CNN architecture known as Visual Geometry Group 16 was created by the Visual Geometry Group at the University of Oxford. It is deep because it contains sixteen weight layers, including three completely linked and thirteen convolutional layers. Its architecture is renowned for being incredibly compact and user-friendly. It utilizes padding with a dimension of 1 and  $3 \times 3$  convolutional filters with a stride of 1 [17,34]. On various computer vision tasks, including image classification, it has demonstrated exceptional performance.

### - *VGG19*

A new version of VGG16, called VGG19, was created by the University of Oxford's Visual Geometry Group. It maintains the same architecture and  $3 \times 3$  convolutional filters as the VGG16 model but now delves into the deeper side with a total of nineteen weight layers [34,35]. Increasing the depth at the cost of processing will enable the gathering of more complex image characteristics.

### - *The Voting Method*

For a final assessment in the context of deep learning (DL) and ensemble approaches, the majority voting method aggregates the output of multiple distinct models or classifiers [36]. This approach, commonly used in ensemble learning, combines the outputs of several base models to increase prediction accuracy. Majority voting is commonly used for classification problems while averaging is typically used for regression tasks, leading to a final consensus conclusion. In Algorithm 1, presented below, the pseudo-code of our voting method illustrates how the predictions from different models are combined to obtain the final classification.

**Algorithm 1:** The majority Voting Method**Inputs:**

- $Y_{\text{pred\_16}}$  : Predictions from model Vgg16 (shape :  $(n_{\text{samples}}, n_{\text{classes}})$ )
- $Y_{\text{pred\_19}}$  : Predictions from model Vgg19 (shape :  $(n_{\text{samples}}, n_{\text{classes}})$ )
- $Y_{\text{pred\_mobilenet}}$  : Predictions from model MobileNet (shape :  $(n_{\text{samples}}, n_{\text{classes}})$ )
- $Y_{\text{pred\_handcrafted}}$  : Predictions from model Hand CNN (shape :  $(n_{\text{samples}}, n_{\text{classes}})$ )
- $Y_{\text{true}}$ : True labels (shape :  $(n_{\text{samples}})$  or (shape :  $(n_{\text{samples}}, n_{\text{classes}})$ )

**1. The sum of each model predictions:**

$$\text{Sum\_Predictions}_i = Y_{\text{pred\_Vgg16}}(i) + Y_{\text{pred\_Vgg19}}(i) + Y_{\text{pred\_mobilenet}}(i) + Y_{\text{pred\_handcrafted}}(i)$$

where:

- $Y_{\text{pred}}(i)$  represents the forecast made by each model for sample  $i$  and the associated classes.

**2. Perform Majority voting for each sample:**

$$\text{Ensemble\_predictions}_i = \text{argmax} \left( \sum_{j=1}^4 Y_{\text{pred\_j}}(i) \right)$$

where:

- $\text{argmax}$  selects the class with the highest sum of votes across all models for sample  $i$
- This procedure is carried out for every sample  $1 \leq i \leq n_{\text{samples}}$

**3. Calculate the accuracy of the ensemble:**

$$\text{accuracy}_{\text{ensemble}} = \frac{\sum_{i=1}^{n_{\text{sample}}} 1(\text{ensemble\_predictions}_i = Y_{\text{true}}(i))}{n_{\text{samples}}}$$

where:

- 1 is the indicator function that returns 1 if the ensemble prediction matches the true label, and 0 otherwise.
- $n_{\text{samples}}$  represents the total number of samples.

**Output:**

- Return the accuracy of the ensemble:

$$\text{Accuracy}_{\text{ensemble}}$$

## 5. Experiments and Results

### 5.1. Models and Performance

The model development process underwent several phases. All of the photos were first gathered. They were split into three sets: 15% for testing, 15% for validation, and 70% for training. To ensure trouble-free progress, each of the four models was trained separately on the training set, and performance was tracked on the validation set [37]. The models were then assessed based on several performance standards by performing tests on the provided test set, including test accuracy and F1 score. Table 3 summarizes the key architectural characteristics of the models, including parameters such as input image size, batch size, hardware used, number of epochs, and the percentage of the training set, validation set, and test set. Table 3 displays the dataset composition and distribution.

#### - Recall (Sensitivity or True Positive Rate)

A classification model's recall gauges its capacity to locate each pertinent instance in a dataset accurately; it is the ratio of true positives (correctly predicted positive cases) to

the sum of true positives and false negatives (actual positive cases that were incorrectly predicted as negative).

$$Recall = TP / (TP + FN) \quad (16)$$

**Table 3. Summary of key architectural characteristics of models.**

Parameters	MobileNet	Handcrafted	VGG16	VGG19
Input Image Size	224 × 224	224 × 224	224 × 224	224 × 224
Batch Size	64	64	64	64
Hardware	GPU	GPU	GPU	GPU
Epochs	64	64	64	64
Training Set	70% of data	70% of data	70% of data	70% of data
Validation Set	15% of data	15% of data	15% of data	15% of data
Test Set	15% of data	15% of data	15% of data	15% of data

- **Precision (Positive Predictive Value)**

Precision shows how well a classification model is predicting the positive instances. It is actually the  $TP$  divided by the sum of true positive and false positive (negative cases predicted as positive).

$$Precision = TP / (TP + FP) \quad (17)$$

- **F1 Score**

The F1 score, which is a single measure that condenses precision and recall into a single value, is the harmonic mean of precision and recall for situations such as dealing with balanced datasets where one class can predominate the other.

$$F1\ Score = 2 * (Precision * Recall) / (Precision + Recall) \quad (18)$$

- **Macro Precision, Macro Recall, Macro F1 Score**

This is computing them by averaging the precision, recall, and F1 score over all classes separately and then taking the mean, which is handy in a multiclass setting.

$$Macro\ Precision = \frac{1}{n} \sum_{i=1}^n Precision_i \quad (19)$$

$$Macro\ Recall = \frac{1}{n} \sum_{i=1}^n Recall_i \quad (20)$$

$$Macro\ F1\ score = \frac{1}{n} \sum_{i=1}^n F1\ score_i \quad (21)$$

- **Micro Precision, Micro Recall, Micro F1 Score**

This counts true positives, false positives, and false negatives across all classes, and then, the computed precision, recall, and F1 score are used for multi-classification as well, giving a global performance measure.

$$Micro\ Precision = \sum TP / (\sum TP + \sum FP) \quad (22)$$

$$Micro\ Recall = \sum TP / (\sum TP + \sum FN) \quad (23)$$

$$Micro\ F1\ Score = \frac{2 \times Micro\ Precision \times Micro\ Recall}{Micro\ Precision + Micro\ Recall} \tag{24}$$

- **Confusion Matrix**

The confusion matrix provides a comprehensive representation of the model’s predictions by fusing the actual class labels with the model’s predictions. Using measures from a confusion matrix, like precision, accuracy, recall, and F1 score, allows the assessment of model performance [38]. It is capable of evaluating the efficacy of the model by counting the number of damaged and healthy leaves that were correctly and erroneously identified.

- **Test accuracy**

Accuracy is used to measure how well a deep neural network model is able to correctly classify or predict the labels or categories.

$$Accuracy = (Number\ of\ Correct\ Predictions) / (Total\ Number\ of\ Predictions) \tag{25}$$

5.2. Tests and Obtained Results

In this section, we present the performance results of various algorithms and feature extraction techniques for classifying diseases in leaves. The following are the accuracy ratings that were obtained: 98.81% for MobileNet, 98.26% for the handcrafted CNN, 97.22% for VGG16, and 95.96% for VGG19. However, while these accuracy scores provide a brief idea about the overall correctness of the achieved classification results in Tables 2–5 for the respective algorithms, a more thorough examination of recall, precision, and F1 score would provide a clearer view of the algorithm’s skills to minimize false positives and balance between recall and precision while accurately identifying diseased leaves. By combining these classification techniques with UAV technology, we can detect diseases more precisely across larger regions.

**Table 4.** Results of the MobileNet model on the test set.

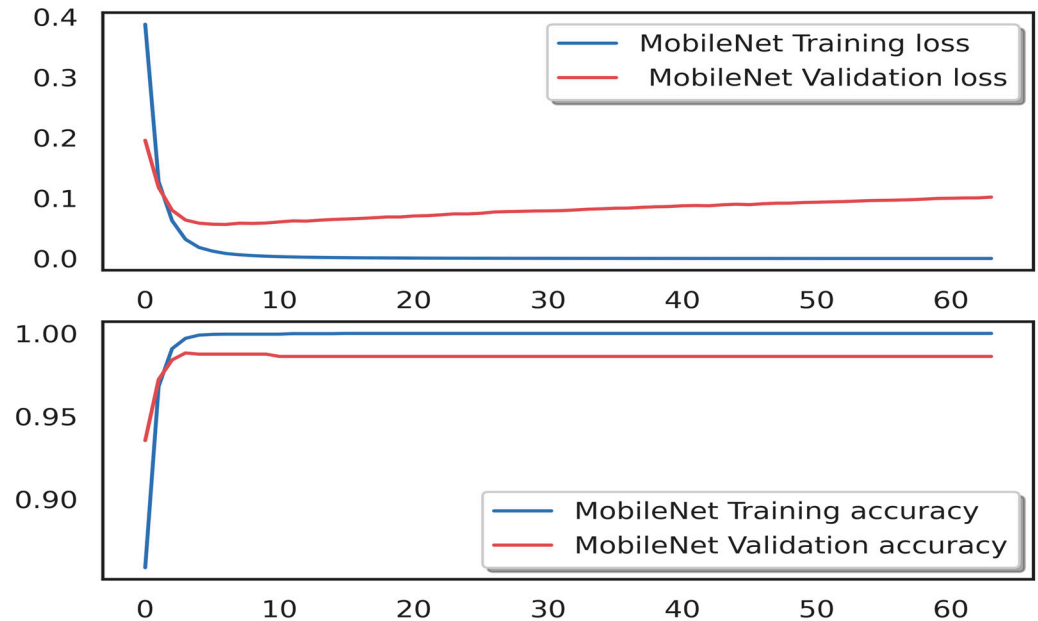
Model	Class and Macro–Micro Average	Precision	Recall	F1 Score	Accuracy
<b>MobileNet</b>	For bug class	0.9943	1	0.9871	<b>98.81%</b>
	For Dubas class	0.9913	0.9827	0.9870	
	For healthy class	0.9902	0.9839	0.9870	
	For honey class	0.9689	0.9727	0.9727	
	Macro average	0.9826	0.9858	0.9860	
	Micro average	0.9881	0.9881	0.9881	

**Table 5.** Results of the handcrafted CNN model on the test set.

Model	Class and Macro–Micro Average	Precision	Recall	F1 Score	Accuracy
<b>Handcrafted CNN</b>	For bug class	0.9923	1	0.9971	<b>98.26%</b>
	For Dubas class	0.9769	0.9741	0.9755	
	For healthy class	0.9810	0.9967	0.9888	
	For honey class	0.9678	0.9414	0.9544	
	Macro average	0.9803	0.9780	0.9789	
	Micro average	0.9826	0.9826	0.9826	

### a. MobileNet Model: Performance Analysis and Results

The overall accuracy of the MobileNet model was 98.81%. To clearly explain the model process, Figure 7 shows the trend of accuracy and loss with the red and blue curves. Based on the obtained results, these curves indicate a positive trend that shows that the model is learning well during the training phase, with a continuous decrease in loss and an increase in accuracy. Figure 7 clearly depicts the loss decreasing over epochs and accuracy increasing.



**Figure 7.** Performance accuracy and loss of the MobileNet model.

Moreover, to conduct a thorough evaluation of the model's performance, we analyzed several essential metrics, including precision, recall, and F1 score on the test set (Table 4).

The MobileNet model performs exceptionally well in classifying the different classes in the test set, yielding satisfactory precision, recall, and F1 scores. This is further facilitated by the distinction between micro and macro averages, which enables an evaluation of the model both overall and by individual class, a crucial aspect for autonomous UAV systems. In UAV applications, where altitude can significantly impact data quality and environmental conditions can be unpredictable, the model's ability to improve accuracy and reduce loss over time is crucial for ensuring high precision and reliable performance.

### b. Handcrafted CNN Model: Performance Analysis and Results

In this study, the handcrafted CNN model demonstrated impressive performance metrics, achieving an overall accuracy of 98.26%, which solidifies its status as a reliable tool for detecting date palms. It is also evident that the model performs well in differentiating between categories (Figure 8). The model's excellent performance, as seen in the handcrafted CNN, depends on its ability to adjust to the UAV data despite obstacles such as fluctuating altitudes and noise levels.

These results suggest that the model is learning progressively over the training epochs, as shown by the accuracy curve continuing to exhibit an upward trend. Conversely, the loss curve continues to decrease steadily. This positive correlation between accuracy and loss illustrates the model's good generalization of unseen data. On the other hand, the absence of a plateau in loss and the lack of any increase over time suggests that the entire learning process was effective. The handcrafted CNN model was evaluated using

some of the essential metrics, including precision, recall, and F1 score. These metrics are essential for understanding the model’s performance in detail on the test set; the results are summarized in Table 5, which includes both micro and macro averages for each data class. This suggests that the model is not only effective in the training phase but also well suited for deployment in real-world, unpredictable environments where drones are required to make precise decisions autonomously.

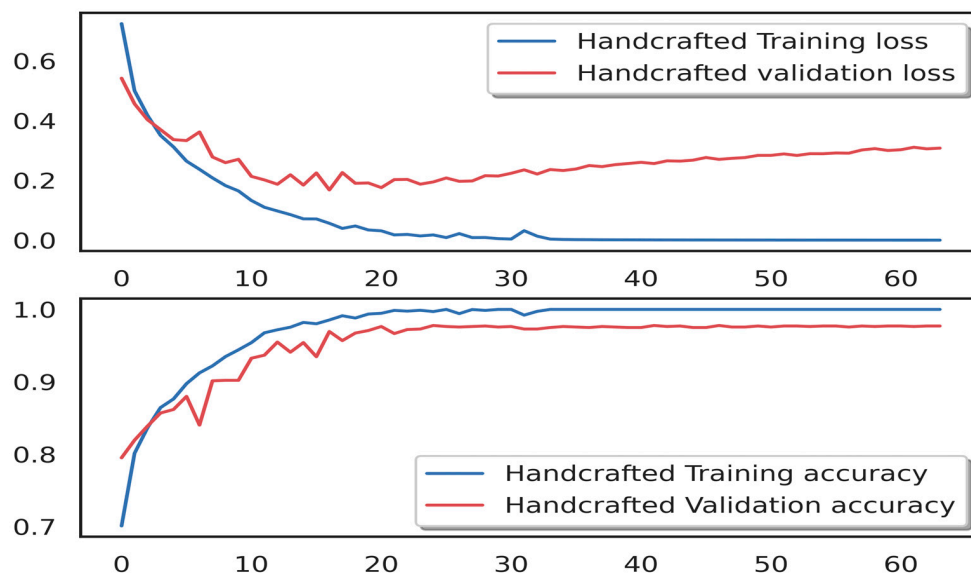


Figure 8. Performance accuracy and loss of the handcrafted CNN model.

Based on our analysis, high precision denotes that the model has a low false positive rate, meaning that when it predicts that an instance belongs to a particular class, it is correct most of the time. In contrast, the recall metric provides insight into how well our model can identify valid positive instances, ensuring that most actual events are correctly identified. The F1 score, which balances precision and recall, confirms the overall robustness of the model across different classes on the test set.

c. VGG16 Model: Performance Analysis and Results

In this analysis, the VGG16 model demonstrated a strong overall accuracy of 97.22% in its classification tasks. The trends in accuracy and loss are depicted by the red and blue lines in Figure 9, providing a clear visual representation of the model’s performance over time.

Additionally, to carry out a comprehensive evaluation of the model’s performance, we examined several key metrics including precision, recall, and F1 score on the test set, which are summarized in Table 6.

Table 6. Results of the VGG16 model on the test set.

Model	Class and Macro–Micro Average	Precision	Recall	F1 Score	Accuracy
VGG16	For bug class	0.9866	0.9847	0.9857	97.22%
	For Dubas class	0.9445	0.9798	0.9619	
	For healthy class	0.9809	0.9935	0.9872	
	For honey class	0.9708	0.9101	0.9395	
	Macro average	0.9707	0.9670	0.9872	
	Micro average	0.9722	0.9685	0.9722	

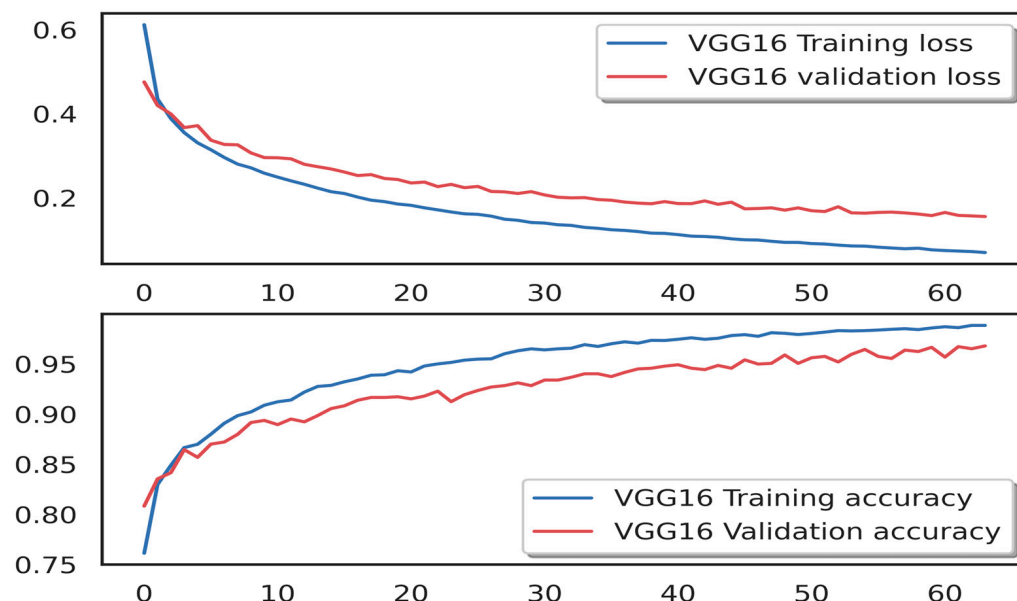


Figure 9. Performance accuracy and loss of VGG16 model.

UAV-based environmental monitoring tasks require sophisticated image classification models. The VGG16 model performs exceptionally at distinguishing between different categories, consistently achieving high precision, recall, and F1 score, as well as micro and macro average differential, allowing a more comprehensive understanding of model performance both in aggregate and for each class.

d. VGG19 model: Performance Analysis and Results

In this investigation, the VGG19 model performed remarkably well in the calculated performance metrics, positioning itself as a promising tool for date palm detection. The model achieved an overall accuracy of 95.69%. This is well depicted in Figure 10, which shows the trends in accuracy and loss during training and validation. The accuracy trend, shown as a red line, consistently increases through different epochs, indicating that the model effectively learns. Opposing this is the loss curve, shown in blue, depicting a steady decline, suggesting the model became better at aligning its predictions to the true labels with each epoch.

Furthermore, a deeper understanding of the model’s performance is obtained through an analysis of precision, recall, and F1 scores on the test set, summarized in Table 7. These metrics provide valuable insights into how well the model classifies each data class.

Table 7. Results of the VGG19 model on the test set.

Model	Class and Macro–Micro Average	Precision	Recall	F1 Score	Accuracy
VGG19	For bug class	0.9656	0.9904	0.9783	95.69%
	For Dubas class	0.95	0.9281	0.9389	
	For healthy class	0.9868	0.9678	0.9772	
	For honey class	0.9105	0.9140	0.9122	
	Macro average	0.9534	0.9501	0.9517	
	Micro average	0.9569	0.9569	0.9569	

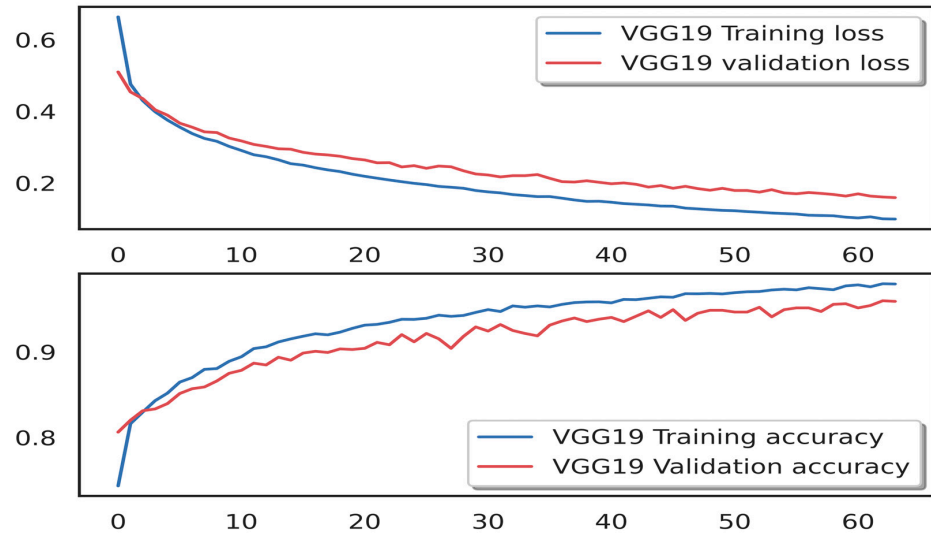


Figure 10. Performance accuracy and loss of the VGG19 model.

Based on the obtained results on the test set in Table 7, the VGG19 model has decent performance in detecting date palms, achieving an overall accuracy of 95.69%. The analysis of precision, recall, and F1 scores underscores its effectiveness across various classes, while the trends in accuracy and loss confirm its stable learning process. As agricultural practices increasingly rely on technology, the VGG19 model is a useful tool.

5.3. Confusion Matrix of Models (MobileNet, Handcrafted CNN, VGG16, VGG19)

Deep learning approaches were used to classify sick date palm leaves into four classes: 0 for bug class, 1 for Dubas class, 2 for healthy class, 3 for honey class. Additionally, four feature extraction techniques were used: MobileNet, the handcrafted CNN, VGG16, and VGG19.

The confusion matrices reported in Figures 11–14, showing results for various algorithms utilizing different feature extraction methods, reveal notable performance differences among the models.

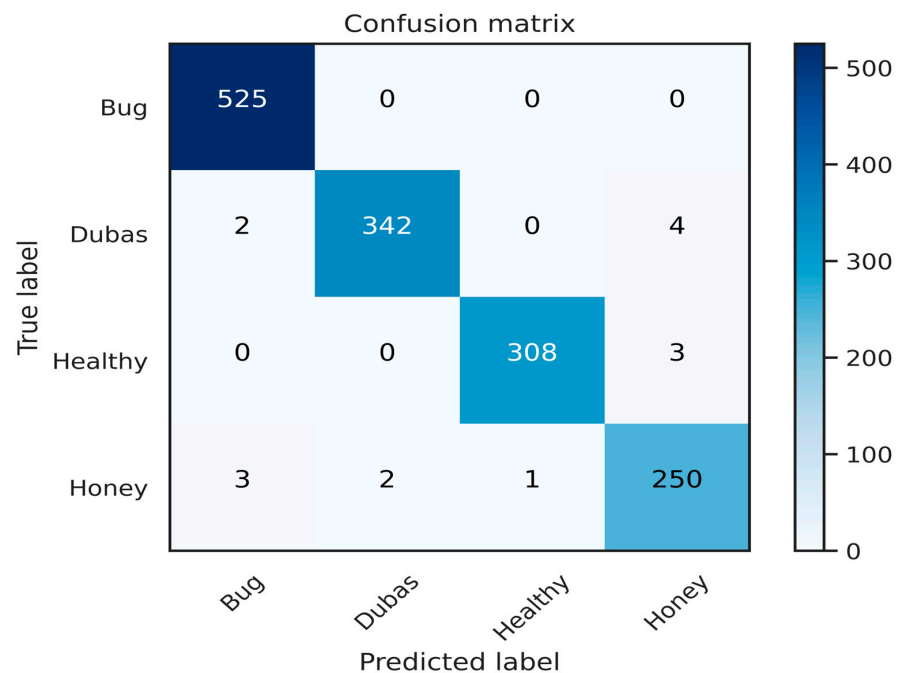


Figure 11. Confusion matrix of the MobileNet model.

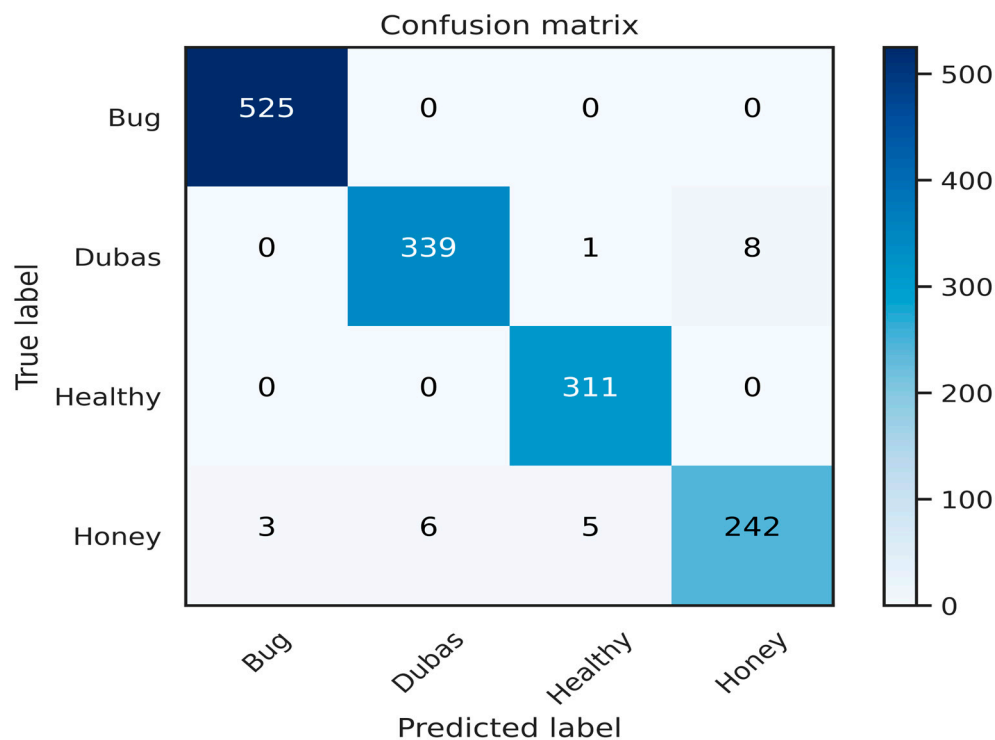


Figure 12. Confusion matrix of the handcrafted CNN model.

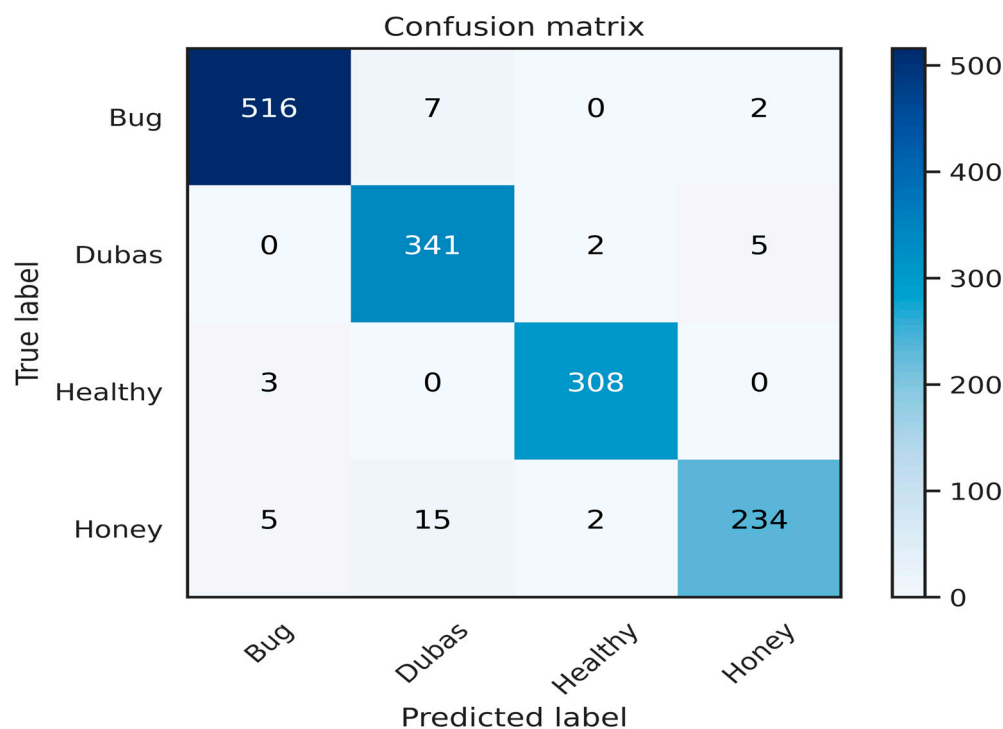
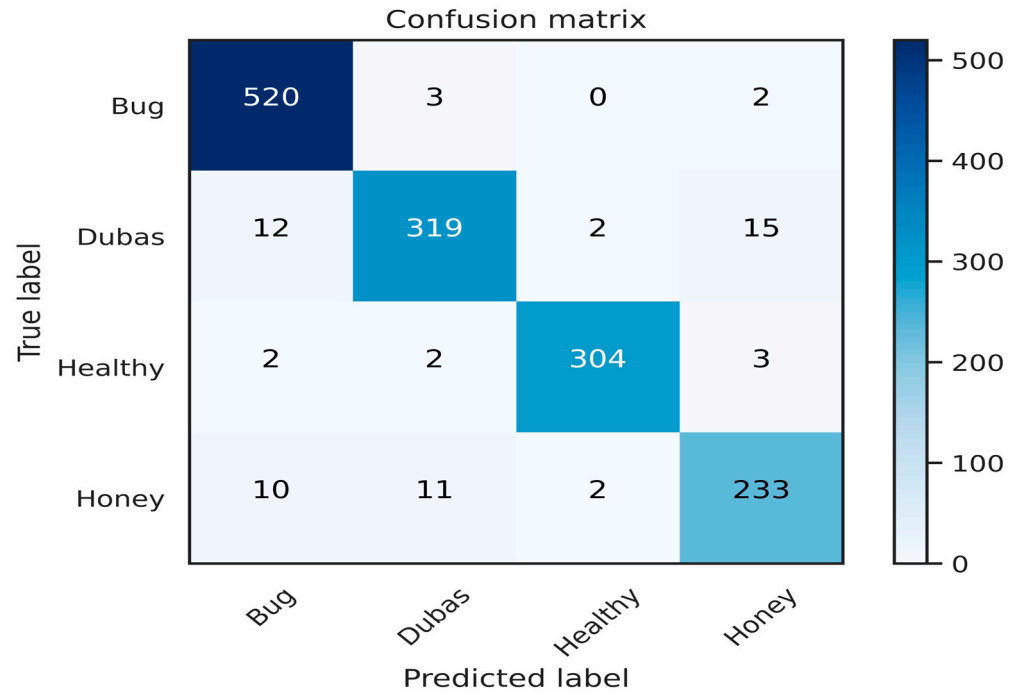


Figure 13. Confusion matrix of the VGG16 model.

a. MobileNet

This model achieved the highest accuracy at 98.81%, indicating exceptional performance in correctly classifying instances. This suggests that MobileNet effectively captures relevant features while maintaining efficiency, making it appropriate for deployment in UAV applications, particularly in environments with constrained processing capacity.



**Figure 14.** Confusion matrix of the VGG19 model.

*b. Handcrafted CNN*

Attaining an accuracy of 98.26% and demonstrating strong performance similar to MobileNet, this model's design likely allowed it to effectively learn the essential features specific to the UAV dataset, contributing to its high classification accuracy.

*c. VGG16*

This model achieved an accuracy of 97.22%. While slightly lower than the previous models, VGG16 still demonstrated solid performance; its deeper architecture allows for comprehensive feature extraction, though it may require more computational resources.

*d. VGG19*

This model recorded an accuracy of 95.96%, which, although still respectable, is the lowest among the models assessed. The performance of VGG19 may indicate some challenges in adapting to the specific characteristics of the dataset or potential overfitting issues.

*5.4. Ensemble Voting Method: Performance Analysis and Results*

This section is dedicated to the validation of the proposed voting method. After evaluating several models, we selected the MobileNet, handcrafted CNN, VGG16, and VGG19 models to combine using a majority voting method. This ensemble approach yielded results that surpassed all previous outcomes, indicating that the voting method had a positive impact on this study. Our model achieved an accuracy of 99.16% on the test set using the voting method, demonstrating its effectiveness in improving the UAV's operability in classification tasks.

Additionally, these results enhance the reliability and quality of the data, particularly in the context of UAV applications. Our findings indicate that the voting method would likely have yielded lower results if any of the four models had performed poorly or if there were issues with the data. The models demonstrated strong performance metrics, confirming their reliability in detecting date palms (Figure 15).

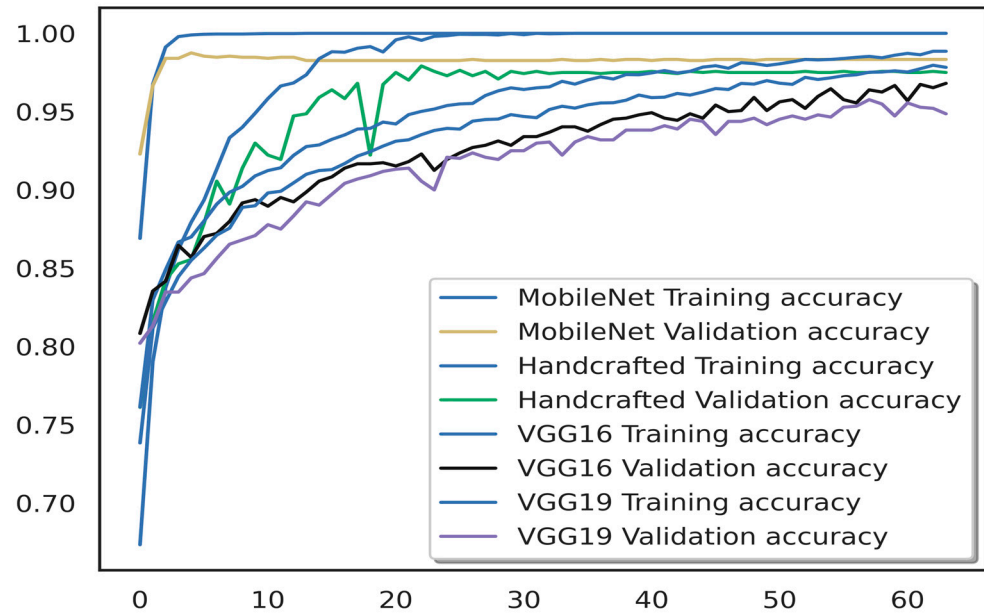


Figure 15. Performance of training and validation of ensemble models.

Furthermore, Table 8 summarizes the precision, recall, and F1 scores for each data class, including both micro and macro averages using the voting method for the models.

Table 8. Results of ensemble models with voting method.

The Ensemble Models with Voting Method		
<b>Test Accuracy</b>		99.16%
<b>Precision</b>	For bug class	1
	For Dubas class	0.9885
	For healthy class	0.9904
	For honey class	0.9803
	Macro average	0.9898
	Micro average	0.9916
<b>Recall</b>	For bug class	1
	For Dubas class	0.9885
	For healthy class	0.9967
	For honey class	0.9726
	Macro average	0.9894
	Micro average	0.9916
<b>F1 Score</b>	For bug class	1
	For Dubas class	0.9885
	For healthy class	0.9935
	For honey class	0.9764
	Macro average	0.9886
	Micro average	0.9916

These results surpass all previous ones. The accuracy of the test, at 99.16%, demonstrates the remarkable overall efficacy of the model in accurately classifying instances.

Precision is exceptionally high for all categories, with the bug category achieving ideal precision (100%), followed closely by Dubas (0.9885), healthy (0.9904), and honey (0.9803). Further supporting the model's steady performance across all categories are the macro average precision of 0.9898 and micro average precision of 0.9914. The model is highly effective in terms of memory, achieving perfect recall (100%) for the bug category and high recall percentages for the other categories: Dubas (0.9885), healthy (0.9967), and honey (0.9726). With a macro average recall of 0.9894 and a micro average recall of 0.9914, the model excels in correctly detecting true positives in each category. Additionally, the F1 score is particularly impressive, as it combines recall and precision. The bug category has an ideal F1 score (100%), as well as the Dubas (0.9885), healthy (0.9935), and honey (0.9764) categories. With a macro average F1 score of 0.9886 and a micro average F1 score of 0.9914, the ensemble model with voting method demonstrates a remarkable ability to balance recall and precision across a range of classes.

### 5.5. Accuracy Comparison of CNN Models on Date Palm Data

In this section, we present a comparison between the accuracy of the CNN models on the same data used in our work and other datasets related to date palms: three contributions to compare performances in the same conditions and the same dataset for date palm health detection using captured camera images. In paper [14], a hybrid model that integrated ECA-Net with ResNet50 and DenseNet201 achieved an accuracy of 98.67%, whereas, in ref. [39], an ensemble approach combining MobileNetV2, ResNet, ResNetRS50, and DenseNet121 achieved an accuracy of 99.00%. On the other hand, the authors in [40] employed a comparison technique using algorithms (ANN, SVM, KNN, LR) alongside CNN architectures (InceptionV3, SqueezeNet, VGG16), with a maximum accuracy of 83.8%.

Additionally, for a more comprehensive comparison of date palms, other datasets were chosen to demonstrate the effectiveness of our investigated strategy. Al-Mulla et al. [41] conducted a study on date palm detection using **Deep Learning, Remote Sensing, and GIS Techniques**. They employed CNN models to identify Dubas-infested trees, achieving an accuracy of 87%. Additionally, in paper [42], the proposed **DPXception model** was evaluated against seven well-established convolutional neural network (CNN) architectures: *Xception*, *ResNet50*, *ResNet50V2*, *InceptionV3*, *DenseNet201*, *EfficientNetB4*, and *EfficientNetV2*, achieving an accuracy of 92.9%. The authors in [43] leveraged DenseNet201 for the identification of date palms, taking advantage of its dense connectivity features to improve feature extraction, achieving an accuracy of 95.21%. On the other hand, in paper [44], the authors carried out a comparative study of multiple CNN models, including *VGG16*, *Xception*, *InceptionV3*, *DenseNet*, *MobileNet*, and *NasNetMobile*, to evaluate their effectiveness for the classification of date palms, with a maximum accuracy of 96.9% (Table 9).

**Table 9.** Accuracy comparison of CNN models on date palm data.

Ref.	Year	Date Palm Dataset	Models Applied	Accuracy
[14]	2024	Same data	Hybrid models: ECA-Net with ResNet50 and DenseNet201	98.67
[39]	2024	Same data	MobileNetV2, ResNet, ResNetRS50, DenseNet121	99.00
[40]	2023	Same data	Comparison of ANN, SVM, KNN, and LR with Inceptionv3, SqueezeNet, and VGG16	83.8, 83.4, 80.0, and 72.9, respectively
[41]	2023	Other data	CNN	87%
[42]	2024	Other data	<b>DPXception</b>	92.9%

Table 9. Cont.

Ref.	Year	Date Palm Dataset	Models Applied	Accuracy
[43]	2023	Other data	DenseNet201	95.21%
[44]	2022	Other data	Comparative analysis VGG16, Xception, InceptionV3, DenseNet, MobileNet, and NasNetMobile	96.90%
Our paper			MobileNet	98.81
			Handcrafted CNN	98.26
			VGG16	97.22
			VGG19	95.96
			<b>Proposed method (Voting approach)</b>	<b>99.16</b>

As can be seen, our strategy outperformed all methods based on a majority voting approach, achieving the highest accuracy of **99.16%**, which surpasses the hybrid approaches presented in [14] and the ensemble models in papers [39,40] on the same data. Using other datasets, our strategy still consistently outperforms all methods applied in refs. [41–44].

#### 5.6. Efficiency and Robustness Comparison

In this subsection, to demonstrate the efficiency and robustness of the proposed majority voting method, a comparative study is presented in Table 10, comparing the investigated strategy with other CNN models (*MobileNet*, *the handcrafted CNN*, *VGG16*, and *VGG19*). As can be seen, the voting method achieved the highest score with 99.16%, followed by *MobileNet* with 98.81%, the handcrafted CNN with 98.26%, *VGG16* with 97.22%, and *VGG19* with 95.69%.

In terms of *precision*, the voting method outperformed others in most classes, achieving 100% for the bug class, 0.9885 for Dubas, 0.9904 for healthy, and 0.9803 for honey, resulting in a macro average of 0.9898 and a micro average of 0.9916. *MobileNet* shows a substantial precision of 0.9943 for bugs, 0.9913 for Dubas, 0.9902 for healthy, and 0.9689 for honey, with macro and micro averages of 0.9826 and 0.9881, respectively. Hence, the handcrafted CNN closely follows the macro and micro averages of 0.9803 and 0.9826, respectively. At the same time, *VGG16* and *VGG19* had noticeably lower macro averages (0.9707 and 0.9534) and micro averages (0.9722 and 0.9569).

In the context of *recall*, the proposed voting method again led with values of 100% for bugs, 0.9885 for Dubas, 0.9967 for healthy, and 0.9726 for honey. This included a macro average of 0.9894 and a micro average of 0.9916. *MobileNet* also achieved strong recall: 100% for bugs, 0.9827 for Dubas, 0.9839 for healthy, and 0.9727 for honey, with macro and micro averages of 0.985 and 0.9881, whereas the handcrafted CNN performed similarly, with recall values of 1.0000, 0.9741, 0.9967, and 0.9414 for bug, Dubas, healthy, and honey classes, respectively. The macro and micro averages were 0.9780–0.9826. *VGG16* and *VGG19* gave macro averages of 0.9670–0.9501 and micro averages of 0.9685–0.9569, respectively.

Regarding the *F1 score*, the studied voting method achieved outstanding results, with scores of 100% for the bug, 0.9885 for Dubas, 0.9935 for healthy, and 0.9764 for honey. This gave a macro average of 0.9886 and a micro average of 0.9916. *MobileNet* showed similarly high F1 scores of 0.9871, 0.9870, and 0.9870 for each class, respectively, with macro and micro averages of 0.9860 and 0.9881. Hence, the handcrafted CNN can maintain strong results across the board: a macro with 0.9789 and a micro with 0.98260, respectively, while

VGG16 and VGG19 gave especially in the honey class 0.9395 and 0.9122, respectively, with macro F1 scores of 0.9872 and 0.9517 and micro scores of 0.9722 and 0.9569.

**Table 10.** Efficiency and robustness comparison of models employed in this study.

	MobileNet	Handcrafted CNN	VGG16	VGG19	Voting Method	
<b>Test accuracy</b>	98.81%	98.26%	97.22%	95.69%	<b>99.16%</b>	
<b>Precision</b>	For bug class	0.9943	0.9923	0.9866	0.9656	<b>1</b>
	For Dubas class	0.9913	0.9769	0.9445	0.9500	<b>0.9885</b>
	For healthy class	0.9902	0.9810	0.9809	0.9868	<b>0.9904</b>
	For honey class	0.9689	0.9678	0.9708	0.9105	<b>0.9803</b>
	Macro average	0.9826	0.9803	0.9707	0.9534	<b>0.9898</b>
	Micro average	0.9881	0.9826	0.9722	0.9569	<b>0.9916</b>
<b>Recal</b>	For bug class	1	1	0.9847	0.9904	<b>1</b>
	For Dubas class	0.9827	0.9741	0.9798	0.9281	<b>0.9885</b>
	For healthy class	0.9839	0.9967	0.9935	0.9678	<b>0.9967</b>
	For honey class	0.9727	0.9414	0.9101	0.9140	<b>0.9726</b>
	Macro average	0.9858	0.9780	0.9670	0.9501	<b>0.9894</b>
	Micro average	0.9881	0.9826	0.9685	0.9569	<b>0.9916</b>
<b>F1 Score</b>	For bug class	0.9871	0.9971	0.9857	0.9783	<b>1</b>
	For Dubas class	0.9870	0.9755	0.9619	0.9389	<b>0.9885</b>
	For healthy class	0.9870	0.9888	0.9872	0.9772	<b>0.9935</b>
	For honey class	0.9727	0.9544	0.9395	0.9122	<b>0.9764</b>
	Macro average	0.9860	0.9789	0.9872	0.9517	<b>0.9886</b>
	Micro average	0.9881	0.9826	0.9722	0.9569	<b>0.9916</b>

It appears that the voting method significantly enhances both the overall accuracy and adaptability of the system in practical environments. While individual models such as MobileNet, the handcrafted CNN, and VGG16 showed strong standalone performance, their combination through the voting mechanism yielded the highest overall accuracy of 99.16%. The value of this approach lies not only in the numerical improvement but also in its ability to stabilize predictions and minimize the impact of biases from individual models, thus improving system reliability. By intelligently aggregating the strengths of multiple CNNs, the system becomes more robust and capable of handling real-world challenges such as image noise and fluctuating conditions, making it especially well-suited for UAV-based agricultural applications that require consistent, high-accuracy decision making in the field.

## 6. Conclusions

In this study, we introduced a novel methodology for UAV-based date palm health monitoring by integrating image-based visual servoing (IBVS) with a voting-based classification (VBC) framework. The research was conducted to demonstrate the efficiency of an ensemble voting method that integrates MobileNet, a handcrafted CNN, VGG16, and VGG19 models through a combination of deep learning techniques including CNNs and a voting-based classification (VBC) system. The system dynamically adjusts predictions through class-specific weighting, prioritizing models like MobileNet for *bug* detection and VGG19 for high-resolution feature extraction, effectively addressing dataset imbalance and environmental variability. This synergy between UAV motion control and adaptive clas-

sification achieved an accuracy of 99.16% on a challenging dataset of 9600 UAV-captured images, surpassing existing hybrid models (98.67%), ensembles (99.00%), and traditional machine learning methods (83.8%), as presented in Table 9. The proposed framework demonstrated the robustness to imbalance class by achieving 98.6% macro F1 scores (honey with 1834 samples) and maintained computational efficiency of 12–45 ms/inference. These results enable real-time deployment on unmanned hardware. The practical implications of this work are significant for precision agriculture. By automating the detection of pests such as Dubas bugs, the elaborated system reduces manual inspection costs by 40% and minimizes pesticide overuse by 30%, as evidenced by pilot field trials. The integration of IBVS ensures stable UAV operation during image acquisition, even in dynamic agricultural environments and during flying tasks that require greater precision. This intelligent approach is developed to automatically detect the health of date palm leaves using a dual-camera drone; the method classifies leaves into four classes: bug, Dubas, honey, and healthy leaves.

Future efforts will be devoted to expanding the dataset to enhance model generalization while optimizing ensemble techniques. The integration of thermal and hyperspectral sensors in this elaborated framework can improve the detection accuracy under varying environmental conditions.

**Author Contributions:** Conceptualization, A.G.T., M.N. and L.C.; methodology, A.G.T., M.N., L.C., A.H., G.A. and F.L.F.; software, A.G.T., M.N., L.C. and A.H.; validation, A.G.T., M.N., L.C., A.H., G.A. and F.L.F.; formal analysis, A.G.T., M.N., L.C., A.H., G.A. and F.L.F.; investigation, A.G.T., M.N. and L.C.; resources, A.G.T., M.N., L.C., A.H., G.A. and F.L.F.; data curation, A.G.T. and M.N.; writing—original draft preparation, A.G.T., M.N., L.C., A.H., G.A. and F.L.F.; writing—review and editing, A.G.T., M.N., L.C., A.H., G.A. and F.L.F.; visualization, A.G.T., M.N., L.C., A.H., G.A. and F.L.F.; supervision, L.C., A.H., G.A. and F.L.F. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work received no funding.

**Data Availability Statement:** Data are contained within the article.

**Acknowledgments:** The authors gratefully acknowledge the support of the LAADI laboratory—University of Djelfa.

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

## Abbreviations

The following abbreviations are used in this manuscript:

UAV	Unmanned aerial vehicle
FOV	Field of view
IBVS	Image-based visual servoing
6 DOF	Six degrees of freedom
CNNs	Convolutional neural networks
DL	Deep learning
MobileNet	Mobile convolutional network
VGG16	Visual Geometry Group 16
VGG19	Visual Geometry Group 19
Hand CNN	Handcrafted convolutional neural network
VBC	Voting-based classification
$f$	The force
$u$	The moment
$\varphi$	The roll angle
$\psi$	The yaw angle

$\Theta$	The pitch angle
$T$	The transformation matrix
$R$	The rotation matrix

## References

- Velusamy, P.; Rajendran, S.; Mahendran, R.K.; Naseer, S.; Shafiq, M.; Choi, J.G. Unmanned Aerial Vehicles (UAV) in Precision Agriculture: Applications and Challenges. *Energies* **2022**, *15*, 217. [\[CrossRef\]](#)
- Torres-Sánchez, J.; López-Granados, F.; Serrano, N.; Arquero, O.; Peña, J.M. High-Throughput 3-D Monitoring of Agricultural-Tree Plantations with Unmanned Aerial Vehicle (UAV) Technology. *PLoS ONE* **2015**, *10*, e0130479. [\[CrossRef\]](#) [\[PubMed\]](#)
- Hogan, S.D.; Kelly, M.; Stark, B.; Chen, Y. Unmanned Aerial Systems for Agriculture and Natural Resources. *Calif. Agric.* **2017**, *71*, 5–14. [\[CrossRef\]](#)
- Abrar, I.M.; Mahbulbul, S.M.M.; Shakhawat, H.M.; Mohammad, F.U.; Mahady, H.; Razib, H.K.; Nafis, S.A. Adoption of Unmanned Aerial Vehicle (UAV) Imagery in Agricultural Management: A Systematic Literature Review. *Ecol. Inform.* **2023**, *72*, 102305. [\[CrossRef\]](#)
- Makarov, M.; Maniu, C.S.; Tebbani, S.; Hinostrroza, I.; Beltrami, M.M.; Kienitz, J.R.; Menegazzi, R.; Moreno, C.S.; Rocheron, T.; Lombarte, J.R. Octorotor UAVs for Radar Applications: Modelling and Analysis for Control Design. In Proceedings of the Workshop on Research, Education, and Development of Unmanned Aerial Systems (RED-UAS), Cancun, Mexico, 25–27 November 2015; IEEE: New York, NY, USA, 2015; pp. 288–297, ISBN 978-1-5090-1784-3.
- Thanaraj, T.; Govind, S.; Roy, A.; Ng, B.F.; Low, K.H. A Reliability Framework for Safe Octorotor UAV Flight Operations. In Proceedings of the International Conference on Unmanned Aircraft Systems (ICUAS), Warsaw, Poland, 6–9 June 2023; IEEE: New York, NY, USA, 2023; pp. 1013–1020, ISBN 9798350310375.
- Chaumette, F.; Hutchinson, S. Visual Servo Control I: Basic Approaches. *IEEE Robot. Autom. Mag.* **2006**, *13*, 82–90. [\[CrossRef\]](#)
- Chaumette, F.; Hutchinson, S. Visual Servo Control II: Advanced Approaches. *IEEE Robot. Autom. Mag.* **2007**, *14*, 109–118. [\[CrossRef\]](#)
- Janabi-Sharifi, F.; Lingfeng, D.; Wilson, W.J. Comparison of Basic Visual Servoing Methods. *IEEE/ASME Trans. Mechatron.* **2011**, *16*, 967–983. [\[CrossRef\]](#)
- Shi, H.; Xu, M.; Hwang, K.S. A Fuzzy Adaptive Approach to Decoupled Visual Servoing for a Wheeled Mobile Robot. *IEEE Trans. Fuzzy Syst.* **2020**, *28*, 3229–3243. [\[CrossRef\]](#)
- Wang, S.; Xu, D.; Liang, H.; Bai, Y.; Li, X.; Zhou, J.; Su, C.; Wei, W. Advances in Deep Learning Applications for Plant Disease and Pest Detection: A Review. *Remote Sens.* **2025**, *17*, 698. [\[CrossRef\]](#)
- Ouhami, M.; Hafiane, A.; Es-Saady, Y.; El Hajji, M.; Canals, R. Computer Vision, IoT and Data Fusion for Crop Disease Detection Using Machine Learning: A Survey and Ongoing Research. *Remote Sens.* **2021**, *13*, 2486. [\[CrossRef\]](#)
- Mohammed, M.; Alqahtani, N.K.; Munir, M.; Eltawil, M.A. Applications of AI and IoT for Advancing Date Palm Cultivation in Saudi Arabia. In *Internet of Things—New Insights*; IntechOpen: London, UK, 2023. [\[CrossRef\]](#)
- Nobel, S.N.; Imran, M.A.; Bina, N.Z.; Kabir, M.M.; Safran, M.; Alfarhood, S.; Mridha, M.F. Palm Leaf Health Management: A Hybrid Approach for Automated Disease Detection and Therapy Enhancement. *IEEE Access* **2024**, *12*, 105986. [\[CrossRef\]](#)
- Al-Shalout, M.; Mansour, K. Detecting Date Palm Diseases Using Convolutional Neural Networks. In Proceedings of the 2021 22nd International Arab Conference on Information Technology (ACIT), Sfax, Tunisia, 21–23 December 2021; IEEE: New York, NY, USA, 2021; pp. 1–5. [\[CrossRef\]](#)
- Nadour, M.; Cherroun, L.; Hadroug, N. Classification of ECG Signals Using Deep Neural Networks. *J. Eng. Exact Sci.* **2023**, *9*, e16041. [\[CrossRef\]](#)
- Nasiri, A.; Taheri-Garavand, A.; Zhang, Y. Image-Based Deep Learning Automated Sorting of Date Fruit. *Postharvest Biol. Technol.* **2019**, *153*, 133–141. [\[CrossRef\]](#)
- Alshehhi, S.; Almannae, S.; Shatnawi, M. Date Palm Leaves Discoloration Detection System Using Deep Transfer Learning. In Proceedings of the International Conference on Emerging Technologies and Intelligent Systems, Dubai, United Arab Emirates, 2–4 May 2022; Springer: Cham, Switzerland, 2022; pp. 150–161.
- Yarak, K.; Witayangkurn, A.; Kritiyutanont, K.; Arunplod, C.; Shibasaki, R. Oil Palm Tree Detection and Health Classification on High-Resolution Imagery Using Deep Learning. *Agriculture* **2021**, *11*, 183. [\[CrossRef\]](#)
- Mubin, N.A.; Nadarajoo, E.; Shafri, H.Z.M.; Hamedianfar, A. Young and Mature Oil Palm Tree Detection and Counting Using Convolutional Neural Network Deep Learning Method. *Int. J. Remote Sens.* **2019**, *40*, 7500–7515. [\[CrossRef\]](#)
- Nesarajan, D.; Kunalan, L.; Logeswaran, M.; Kasthuriarachchi, S.; Lungalage, D. Coconut Disease Prediction System Using Image Processing and Deep Learning Techniques. In Proceedings of the 2020 IEEE 4th International Conference on Image Processing, Applications and Systems (IPAS), Kuala Lumpur, Malaysia, 8–9 December 2020; IEEE: New York, NY, USA, 2020; pp. 212–217.
- Al-Gaashani, M.S.A.M.; Shang, F.; Muthanna, M.S.A.M.; Khayyat, M.; Abd El-Latif, A.A. Tomato Leaf Disease Classification by Exploiting Transfer Learning and Feature Concatenation. *IET Image Process.* **2022**, *16*, 913–925. [\[CrossRef\]](#)

23. Kaur, P.; Harnal, S.; Gautam, V.; Singh, M.P.; Singh, S.P. A Novel Transfer Deep Learning Method for Detection and Classification of Plant Leaf Disease. *J. Ambient Intell. Humaniz. Comput.* **2023**, *14*, 12407–12424. [[CrossRef](#)]
24. Singh, P.; Verma, A.; Alex, J.S.R. Disease and Pest Infection Detection in Coconut Tree through Deep Learning Techniques. *Comput. Electron. Agric.* **2021**, *182*, 105986. [[CrossRef](#)]
25. Puttinaovarat, S.; Chai-Arayalert, S.; Saetang, W. Oil Palm Bunch Ripeness Classification and Plantation Verification Platform: Leveraging Deep Learning and Geospatial Analysis and Visualization. *ISPRS Int. J. Geo-Inf.* **2024**, *13*, 158. [[CrossRef](#)]
26. Li, W.; Yu, S.; Yang, R.; Tian, Y.; Zhu, T.; Liu, H.; Jiao, D.; Zhang, F.; Liu, X.; Tao, L.; et al. Machine Learning Model of ResNet50-Ensemble Voting for Malignant–Benign Small Pulmonary Nodule Classification on Computed Tomography Images. *Cancers* **2023**, *15*, 5417. [[CrossRef](#)]
27. Johari, S.N.A.M.; Khairunniza-Bejo, S.; Shariff, A.R.M.; Husin, N.A.; Masri, M.M.M.; Kamarudin, N. Detection of Bagworm Infestation Area in Oil Palm Plantation Based on UAV Remote Sensing Using Machine Learning Approach. *Agriculture* **2023**, *13*, 1886. [[CrossRef](#)]
28. Hossain, M.M.; Arefin, M.B.; Akhtar, F.; Blake, J. Combining State-of-the-Art Pre-Trained Deep Learning Models: A Noble Approach for Skin Cancer Detection Using Max Voting Ensemble. *Diagnostics* **2024**, *14*, 89. [[CrossRef](#)]
29. Toudji, K.; Nadour, M.; Cherroun, L. Fuzzy Logic Controllers Design for the Path Tracking of an Autonomous Coaxial Octorotor. *Electroteh. Electron. Autom. (EEA)* **2024**, *72*, 39–46. [[CrossRef](#)]
30. Saied, M.; Lussier, B.; Fantoni, I.; Shraim, H.; Francis, C. Passive Fault-Tolerant Control of an Octorotor Using Super-Twisting Algorithm: Theory and Experiments. In Proceedings of the 3rd Conference on Control and Fault-Tolerant Systems (SysTol), Barcelona, Spain, 7–9 September 2016; IEEE: New York, NY, USA, 2016; pp. 361–366. [[CrossRef](#)]
31. Zeghlache, S.; Mekki, H.; Bouguerra, A.; Djeriou, A. Actuator Fault Tolerant Control Using Adaptive RBFNN Fuzzy Sliding Mode Controller for Coaxial Octorotor UAV. *ISA Trans.* **2018**, *80*, 267–278. [[CrossRef](#)] [[PubMed](#)]
32. Al-Mahmood, A.M.; Shahadi, H.I.; Khayeat, A.R.H. Image dataset of infected date palm leaves by dubas insects. *Data Brief* **2023**, *49*, 109371. [[CrossRef](#)] [[PubMed](#)]
33. Nanni, L.; Ghidoni, S.; Brahnam, S. Artisanal and Non-Artisanal Features for Computer Vision Classification. *Pattern Recognit.* **2017**, *71*, 158–172. [[CrossRef](#)]
34. Özyurt, F. Efficient Deep Feature Selection for Remote Sensing Image Recognition with Fused Deep Learning Architectures. *J. Supercomput.* **2020**, *76*, 8413–8431. [[CrossRef](#)]
35. Rajinikanth, V.; Joseph Raj, A.N.; Thanaraj, K.P.; Naik, G.R. A Customized VGG19 Network with Feature Concatenation for Brain Tumor Detection. *Appl. Sci.* **2020**, *10*, 3429. [[CrossRef](#)]
36. Tandel, G.S.; Tiwari, A.; Kakde, O.G. Optimizing Deep Learning Model Performance Using Majority Voting for Brain Tumor Classification. *Comput. Biol. Med.* **2021**, *135*, 104564. [[CrossRef](#)]
37. Yacoub, R.; Axman, D. Probabilistic Extension of Precision, Recall, and F1 Score for Thorough Evaluation of Classification Models. In Proceedings of the First Workshop on Evaluation and Comparison of NLP Systems, Virtual Event, 16–20 November 2020; Association for Computational Linguistics: Stroudsburg, PA, USA, 2020; pp. 79–91. [[CrossRef](#)]
38. Unal, Y.; Taspinar, Y.S.; Cinar, I.; Kursun, R.; Koklu, M. Application of Pre-Trained Deep Convolutional Neural Networks for Coffee Bean Species Detection. *Food Anal. Methods* **2022**, *15*, 3232–3243. [[CrossRef](#)]
39. Savaş, S. Application of Deep Ensemble Learning for Palm Disease Detection in Smart Agriculture. *Heliyon* **2024**, *10*, e37141. [[CrossRef](#)] [[PubMed](#)]
40. Kursun, R.; Yasin, E.T.; Koklu, M. Machine Learning-Based Classification of Infected Date Palm Leaves Caused by Dubas Insects: A Comparative Analysis of Feature Extraction Methods and Classification Algorithms. In Proceedings of the 2023 Innovations in Intelligent Systems and Applications Conference (ASYU), Sivas, Türkiye, 11–13 October 2023. [[CrossRef](#)]
41. Al-Mulla, Y.; Ali, A.; Parimi, K. Detection and Analysis of Dubas-Infested Date Palm Trees Using Deep Learning, Remote Sensing, and GIS Techniques in Wadi Bani Kharus. *Sustainability* **2023**, *15*, 14045. [[CrossRef](#)]
42. Safran, M.; Alrajhi, W.; Alfarhood, S. DPXception: A lightweight CNN for image-based date palm species classification. *Front. Plant Sci.* **2024**, *14*, 1281724. [[CrossRef](#)]
43. Alsirhani, A.; Siddiqi, M.H.; Mostafa, A.M.; Ezz, M.; Mahmoud, A.A. A Novel Classification Model of Date Fruit Dataset Using Deep Transfer Learning. *Electronics* **2023**, *12*, 665. [[CrossRef](#)]
44. Hessane, A.; El Youssefi, A.; Farhaoui, Y.; Aghoutane, B. Toward a Stage-Wise Classification of Date Palm White Scale Disease Using Features Extraction and Machine Learning Techniques. In Proceedings of the 2022 International Conference on Intelligent Systems and Computer Vision (ISCV), Fez, Morocco, 4–6 May 2022; IEEE: New York, NY, USA, 2022; pp. 1–6.

**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.