

Segmentation of Cactus Diseases Using Machine and Deep Learning)

AISSAM EL IBRAHIMI, NABIL EL AKCHIOUI

Laboratory of R&D in Engineering Sciences, Department Physics
Faculty of Sciences and Techniques Al-Hoceima, Abdelmalek Essaadi University
Tétouan, 2117
MOROCCO

Abstract: - Artificial intelligence and machine learning play a critical role in plant disease detection and management. This study proposes an approach for diagnosing and segmenting cactus diseases using Random Forest (RF) and compares its performance with SVM, ANN, KNN, CNN, and U-Net models. A dataset of 20 cactus images (10 cochineal and 10 black bacterial soft rot), resized to 720×720 pixels, was used. Experimental results show that U-Net achieved the highest accuracy, reaching 98% for cochineal and 94% for black bacterial soft rot. RF also demonstrated strong performance, achieving 92% and 91%, respectively, outperforming several traditional machine learning models. These results confirm the effectiveness of U-Net for accurate segmentation and the suitability of RF for reliable disease detection with limited datasets, supporting their application in precision agriculture.

Key-Words: - Image processing, Image segmentation, Machine learning, Deep learning, Cactus disease, Agriculture, Plant disease.

Received: March 29, 2026. Revised: May 23, 2026. Accepted: June 27, 2026. Published: September 21, 2026.

1 Introduction

In recent years, artificial intelligence and machine learning have revolutionized various fields, including agriculture, [1]. These advances have significantly improved techniques for detecting and managing plant diseases, which are essential for crop health and productivity, [2]. Among the different machine learning algorithms, Random Forest has established itself as a robust and efficient method for managing complex classification and segmentation tasks.

This study presents a new machine learning approach for detecting and segmenting cactus diseases using the Random Forest algorithm. Cacti are vital plants in arid regions, and their health is essential to ecological and economic balance. However, detecting diseases in cacti presents unique challenges due to the strong visual similarity between healthy and diseased regions, as well as the complex textures and shapes of the plants.

Existing research on plant disease detection often relies on deep learning algorithms, such as convolutional neural networks (CNNs), which have shown remarkable performance in image classification and object recognition tasks, [3]. However, CNNs generally require large datasets to achieve high accuracy and are prone to overfitting when trained on small datasets. Traditional machine learning methods, such as support vector machines (SVMs), artificial neural networks (ANNs), and k-nearest neighbors (k-NNs), have also been used, but they often struggle with large and noisy datasets.

On the other hand, the Random Forest algorithm offers several advantages, including robustness against overfitting, the ability to handle noisy data,

and efficiency with smaller datasets. This makes it particularly suitable for our task, where the image dataset on cactus disease is of limited size, [4].

In this study, we compiled a comprehensive set of cactus images, capturing a range of diseases. The data set poses significant challenges due to the subtle differences between healthy and diseased regions, requiring a method to effectively discern these nuances. Our Random Forest approach is designed to address these challenges by leveraging its holistic learning capabilities to improve classification accuracy and segmentation performance.

The segmentation process begins with converting images to grayscale and applying edge detection filters (Roberts, Sobel, Scharr, Prewitt) to highlight diseased boundaries. Texture and statistical filters (Gaussian, median, variance) capture differences between healthy and infected tissues. These extracted features train a Random Forest model, which classifies pixels into healthy or diseased regions.

This approach ensures precise disease localization, validated against manually annotated ground truth, and demonstrates high reliability. Results show that the Random Forest model achieves strong performance, confirming its potential in precision agriculture and plant disease management. Several works and surveys for the Detection and Segmentation of plant diseases have been recently published, [5], [6], [7]. On the other hand, earlier work in this field relied on traditional machine learning and statistical classifiers (for example, Support Vector Machine (SVM), Artificial Neural Network (ANN), Naive Bayes, Nearest Neighbors, and so on).

Through extensive experiments, we demonstrate

that our method achieves high precision in detecting and segmenting cactus diseases. Moreover, our approach is effective even with relatively small datasets, thus meeting a common limitation in many machine learning and deep learning methodologies.

The contributions of this work are threefold: (1) Introduce a new application of ML algorithms and CNN (U-net) for the detection and segmentation of cactus diseases. (2) Compile a challenging dataset of cactus disease images. (3) Demonstrate the effectiveness of our method to overcome the limitations of small datasets and achieve high accuracy in disease detection and segmentation.

In the section Related Work, previous work done by several researchers in the field of cactus disease detection using artificial intelligence is discussed. Section 2: Proposed Approach and Methods for ML and DL. This section provides a detailed description of the proposed method for detecting cactus diseases, including the following steps: Data Preparation, Image Preprocessing, Feature Extraction, Model Training, Model Evaluation, and Segmentation of New Images. Section 3: Experiments and Results. This section reports the experiments conducted following the proposed method and the results obtained from these experiments. Section 4: Conclusion and Future Works. In conclusion, the main findings of this study are summarized, their implications are discussed, and avenues for future work in this area are suggested.

2 Related work

Agriculture forms the fundamental pillar of nations. Research in this field aims to improve the quality and quantity of products while reducing costs and increasing profits. However, the quality of agricultural products can be compromised by plant diseases caused by various pathogens, such as fungi, bacteria, and viruses. Consequently, early detection and classification of plant diseases are essential tasks. Farmers require constant monitoring by experts, which can be costly and time-consuming. Thus, many systems have been proposed to address, or at least mitigate, these issues using image processing techniques and automated classification tools, [8],[9].

Image processing and artificial intelligence techniques were employed to diagnose grapevine leaf diseases, [10], focusing primarily on downy and powdery mildew. Background regions were removed using masking techniques to improve classification accuracy. To preserve information related to the affected regions, anisotropic diffusion was applied. Image segmentation was performed using the k-means clustering algorithm, followed by feature extraction based on the gray-level co-occurrence

matrix (GLCM). Finally, disease classification was carried out using a feed-forward neural network.

A support vector machine (SVM)-based approach was proposed for the detection and classification of rose leaf diseases, [11], including black spot and anthracnose. Image segmentation was achieved through thresholding using Otsu's method to determine optimal threshold values. Feature extraction involved discrete wavelet transform (DWT)-based features, discrete cosine transform (DCT) coefficients, and eleven Haralick texture features, resulting in effective classification accuracy.

Methods for detecting and classifying apple fruit diseases, such as canker, apple rot, and apple spots, were investigated, [12]. The authors employed k-means clustering for image segmentation, followed by feature extraction and classification using a multi-class support vector machine.

A technique was developed to detect tomato leaf diseases, including late blight and early blight, [13]. Image preprocessing involved several enhancement operations, such as smoothing, noise reduction, and image isolation. Feature extraction was performed using the Gabor wavelet transform, while support vector machine (SVM) classification employed Cauchy, Laplacian, and inverse multiquadric kernels.

Image preprocessing involved several enhancement operations, [14]. The main stages of image processing for plant disease detection and classification, namely image acquisition and preprocessing segmentation, feature extraction, and classification. Techniques such as Otsu's thresholding method, RGB-to-HSI color space conversion, and k-means clustering were investigated. Among these, k-means clustering yielded the most accurate results.

Neural network-based techniques for cotton leaf disease classification, [15]. K-means clustering was employed for image segmentation to analyze various cotton leaf diseases, including red and white spots. The proposed approach achieved a recognition accuracy of 89.56% with a processing time of 436.95 seconds.

Several studies have addressed the problem of leaf disease detection. A potato leaf disease detection system based on image processing and machine learning was developed, achieving an accuracy of 95% on a dataset of 300 images, [16]. A cotton leaf disease detection system using artificial neural networks was proposed, reporting a classification accuracy of 80%, [17].

System for classifying healthy and diseased regions of rice leaves, [18]. achieving an accuracy exceeding 90% using a backpropagation neural network, [19]. employed a modified convolutional neural network (CNN) to detect tomato leaf diseases,

achieving an average accuracy of 94–95%.

A system was introduced to assess the severity of potato diseases using image processing techniques combined with neural networks, [20]. achieving an accuracy of 93%. Image processing with adaptive thresholding techniques was combined to detect potato diseases, [21]. reporting an accuracy of 96%.

A texture analysis-based approach for grapevine leaf disease identification was developed, achieving an accuracy of 96.6%, [22]. Additional studies have explored various techniques for leaf disease detection and classification, reporting promising results with accuracies reaching up to 98.74%, [23], [24].

3 Proposed Method

In this study, we propose a cactus disease detection framework based on two main approaches: machine learning (ML) and deep learning (DL). The ML-based approach is primarily employed for disease segmentation by combining image processing techniques with classification methods to isolate and identify diseased regions, as illustrated in Figure 1 (Appendix) illustrates the overall workflow of the proposed methodology for cactus disease segmentation. The process begins with image acquisition, followed by a data preparation stage that includes background removal, uniform brightness reduction to compensate for illumination variations, and image resizing to ensure consistent input dimensions. Subsequently, discriminative features related to color, texture, and edge information are extracted from the preprocessed images. The extracted features are then divided into training and testing subsets. During the model training phase, the classifier is employed to learn the mapping between the extracted features and the corresponding class labels, resulting in a trained classification model. This trained model is subsequently used for performance evaluation as well as for the segmentation of new, unseen cactus images after applying the same preprocessing steps. The final output is a pixel-level segmentation map highlighting diseased regions, [25].

In parallel, the DL-based approach explores the use of deep learning architectures, specifically convolutional neural networks (CNNs) and U-Net, to achieve improved segmentation accuracy. This dual-strategy framework facilitates a comprehensive comparison of the strengths and limitations of machine learning and deep learning models in the context of cactus disease detection.

3.1 Machine Learning Approach

illustrates the overall workflow of the proposed methodology for cactus disease segmentation. The process begins with image acquisition, followed by

a data preparation stage that includes background removal, uniform brightness reduction to compensate for illumination variations, and image resizing to ensure consistent input dimensions. Subsequently, discriminative features related to color, texture, and edge information are extracted from the preprocessed images. The extracted features are then divided into training and testing subsets. During the model training phase, the classifier is employed to learn the mapping between the extracted features and the corresponding class labels, resulting in a trained classification model. This trained model is subsequently used for performance evaluation as well as for the segmentation of new, unseen cactus images after applying the same preprocessing steps. The final output is a pixel-level segmentation map highlighting diseased regions.

3.1.1 Image acquisition

Images of cactus leaf diseases were captured using an iPhone 11 Pro Max equipped with a 12 MP camera (f/1.8 aperture, PDAF, and OIS). Cactus diseases vary according to seasonal factors, humidity, temperature, and light conditions, and may be caused by pathogens such as bacteria or scale insects, resulting in symptoms ranging from damage to young shoots to severe tissue rot.

To characterize the incidence of the two selected diseases, images were collected from real cultivation fields in the Tangier–Tetouan–Al Hoceima region of Morocco under diverse environmental and weather conditions, thereby ensuring dataset robustness. A total of 1,192 images were acquired and categorized into three classes: healthy cactus, mealybugs cochineal, and bacterial black soft rot. These diseases were selected due to their high prevalence in the region and their significant economic impact.

The dataset contains two different disease types and healthy cactus, for a total of three types: mealybugs cochineal, bacterial black soft rot and healthy cactus, which are represented in Figure 1. These two cactus diseases were selected because they are frequently observed in the cactus growing region of Tangier-Tetouan-Al Hoceima in Morocco, and can cause significant economic losses. Lesions caused by the same disease show similarities under similar natural conditions. Figure 1 shows representative images of the two common cactus diseases and healthy cactus. It can be seen that the two common diseases have distinctly recognizable visual characteristics. Mealybugs appear as white, cotton-like masses on cactus surfaces, while bacterial black soft rot manifests as black, soft lesions, clearly distinguishable from other infections. Thus, cactus diseases can be classified by their distinct visual characteristics. All images were saved in RGB color

space and stored in uncompressed JPG format.

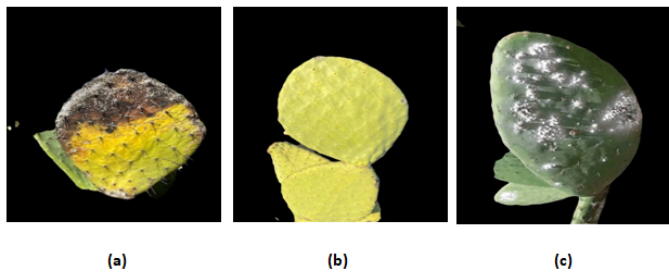


Fig. 1: Two cactus diseases:(a) Cactus with bacterial black soft rot.(b) Healthy cactus. (c) Cactus with mealybugs.

Source: Created by the authors.

3.1.2 Image pre-processing

To speed up the training process and achieve more realistic modeling of the tests, the images in the dataset are resized from their original size (720 x 1280) to resolutions of (720 x 720). This process of optimizing input or output variables reduces the processing time of the drive while preserving data integrity and minimizing the loss of information in the image. In addition, the brightness of the photos has been reduced to compensate for the intensity of the sun. The PhotoRoom app is a popular tool for preprocessing images, including background removal, cropping, and visual enhancement. A uniform brightness reduction was applied to all cactus images in the dataset to ensure consistent illumination conditions and to mitigate the effects of strong sunlight during image acquisition. This preprocessing step reduces unwanted lighting variations and prevents the introduction of bias between healthy and diseased classes. Applying identical brightness settings to all images also guarantees reproducibility and stability in the subsequent feature extraction and segmentation processes.

3.1.3 Feature Extraction

Feature extraction is a crucial step in image processing and pattern recognition, especially for plant disease detection and segmentation applications. In this study, we focus on extracting relevant features from cactus images in order to accurately detect and segment diseases. These features provide essential information about contours, textures, and color variations in images, which are used to form the ML models. Our feature extraction method combines several image analysis techniques to capture the details needed to classify pixels into healthy or diseased areas. I used a labeled image as

the output information. The Table 1 (Appendix) expresses all the characteristic data. The main stages of feature extraction include:

Grayscale Conversion Grayscale conversion is an initial and essential step in simplifying images for feature extraction, especially in applications like plant disease detection, where texture and edge details are critical. By reducing a color image to a grayscale image, [26]. we maintain essential information while minimizing the data complexity. This step is beneficial because it helps reduce computational costs without losing crucial visual patterns that contribute to disease detection. In the grayscale conversion process, we calculate a weighted sum of the red (R), green (G), and blue (B) color channels. This weighted approach is based on the human visual system's varying sensitivity to each color, particularly its high sensitivity to green light. The conversion formula is as follows:

$$\text{Gray} = 0.299 \cdot R + 0.587 \cdot G + 0.114 \cdot B \quad (1)$$

R, G B represent the red, green and blue channel intensities respectively. This weighting reflects the sensitivity of the human eye to different colors, giving more weight to green and less to blue.

The grayscale conversion defined in Equation (1) plays a critical role in the effectiveness of gradient-based edge detectors such as Sobel and Scharr. From a mathematical perspective, these operators compute spatial intensity gradients by convolving the grayscale image with directional derivative kernels. Consequently, the magnitude and accuracy of the detected gradients directly depend on the contrast distribution in the grayscale representation. The weighting coefficients in Equation (1) emphasize the green channel, which contributes the largest weight (0.587) due to its higher sensitivity in natural illumination conditions and its dominant presence in vegetation imagery. This emphasis enhances intensity variations between healthy and diseased cactus regions, thereby increasing the gradient magnitude at texture and boundary transitions. As a result, Sobel and Scharr filters benefit from improved edge responses and more stable gradient estimation, particularly in regions with subtle intensity differences and poorly defined boundaries. This preprocessing step therefore supports more accurate texture characterization and boundary detection, which contributes to the improved segmentation performance observed in the experimental results.

Color Channel Extraction Color channel extraction is a fundamental step in image processing and feature extraction, especially for plant disease detection applications, where each color channel (red, green, and blue) may contain distinct information

about the texture and appearance of affected areas, [27]. Once the image is converted to grayscale, it is separated into three distinct color channels: red (R), green (G), and blue (B). This separation allows us to capture specific variations in each channel to detect anomalies or textures linked to diseases. Each channel is then flattened into a one-dimensional array to be used directly as a feature set in the classification or detection model.

Edge Detection Edge detection is a critical step in the feature extraction process, aimed at identifying the contours and boundaries of infected areas within images. By isolating edges, we can better define the regions affected by disease, which helps in distinguishing healthy from diseased tissue based on sharp transitions in pixel intensity. Several edge detection techniques are applied to capture the contours and boundaries of infected areas: Edge Detection Filters (Roberts, Sobel, Scharr, Prewitt), [28]. Edge detection filters are essential tools in image processing to highlight the contours and boundaries of objects in an image. Different types of filters are used, including Roberts, Sobel, Scharr and Prewitt filters, which detect intensity gradients in different directions. Various edge detection filters are used to detect intensity gradients in different orientations, each having unique properties:

Roberts Filter: Detects edges by calculating the gradient between neighboring pixels, making it particularly sensitive to diagonal edges. It works well for detecting fine details but is more prone to noise.

Sobel Filter: Calculates the gradient using larger neighborhoods of pixels, focusing on vertical and horizontal edges. This filter is commonly used because it balances edge sensitivity and noise resistance.

Scharr Filter: A modified version of the Sobel filter, the Scharr filter uses a different convolution matrix to enhance gradient detection, particularly where subtle edge changes occur. It provides finer edge details compared to Sobel.

Prewitt Filter: Similar to Sobel, Prewitt is effective for vertical and horizontal edges but less sensitive to diagonal transitions. This filter is commonly applied for edge detection with moderate noise levels.

Gaussian filtering Gaussian filtering is an image processing technique widely used to reduce noise and smooth images. It is particularly well suited for random noise reduction while preserving important image contours and details. Gaussian filtering, [29], uses a Gaussian distribution function to calculate the weighted average of neighboring pixels around each pixel of the image. This makes it possible to smooth the image by giving more weight to close pixels and less weight to distant pixels. The

convolution kernel used to apply the Gaussian filter is a discrete approximation of the Gaussian distribution function. The mathematical equation of the Gaussian convolution nucleus $G(x,y)$ for a standard deviation σ is given by:

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (2)$$

$I(i, j)$ represents the intensity of the pixel at position (i, j) in the window W . The symbol μ is the mean intensity of the pixels in the window W , and n is the size of the window.

Median filtering Median filtering, [30], is an image processing technique used to reduce noise while preserving image contours and details. Unlike averaging filtering which takes the average values of neighboring pixels, median filtering uses the median value of pixels in a neighborhood to replace the value of the central pixel. For each pixel in the image, median filtering uses a mask (or a window) of specified size to collect values of neighboring pixels. Then, it replaces the value of the central pixel with the median of the values collected in the neighborhood. This is repeated for all pixels in the image.

statistical features Calculation of the Local Variance Local variance calculation is an important technique in image processing to assess local variations in pixel intensity. This makes it possible to differentiate between regions of homogeneous texture and regions affected by diseases, since variations in intensity can reveal subtle details in the image. For each pixel of the image, a window (usually 3x3) is defined around the central pixel. Then, the variance of the pixel intensities in this window is calculated. Variance measures the dispersion of intensity values in the vicinity of the central pixel, which allows the local variation of intensities to be evaluated. The mathematical equation to calculate the local variance in a window W of size $n \times n$ around the pixel (x,y) is given by:

$$\text{Variance}(x, y) = \frac{1}{n^2} \sum_{i=1}^n \sum_{j=1}^n (I(i, j) - \mu)^2 \quad (3)$$

$I(i, j)$ represents the intensity of the pixel at position (i, j) in the window W . μ is the mean of the intensities of the pixels in the window W . n is the size of the window.

Although Table 1 (Appendix) presents individual pixel values and filter responses, these values are not used independently during the classification process. Instead, they are integrated within a broader statistical representation aimed at characterizing the global texture properties of the image. Such properties include contrast, homogeneity, and spatial

distribution of edges and textures, which are derived through edge detection filters (Roberts, Sobel, Scharr, and Prewitt) and smoothing filters (Gaussian and Median). This statistical feature representation enhances the ability of the Random Forest model to discriminate between healthy and diseased cactus regions, particularly in visually similar scenarios. Mathematically, the feature vector associated with a pixel or local region can be expressed as:

$$\mathbf{f} = [R, G, B, \phi_{\text{Roberts}}, \phi_{\text{Sobel}}, \phi_{\text{Median}}, \dots] \quad (4)$$

where $\phi(\cdot)$ denotes the feature extracted using the corresponding operator. ϕ denotes the response of the corresponding filter. These feature vectors are further aggregated within spatial neighborhoods to extract higher-level texture descriptors, allowing the Random Forest classifier to learn non-linear decision boundaries in the feature space. This mathematical formulation explains the robustness of the proposed approach in scenarios with high visual similarity between healthy and diseased regions.

The window size n used in Gaussian filtering and local variance computation plays a crucial role in balancing noise suppression and detail preservation. From a theoretical perspective, increasing the window size enhances the smoothing effect of the Gaussian filter by averaging pixel intensities over a larger neighborhood. This reduces high-frequency noise and illumination variations; however, excessive smoothing may attenuate fine structural details and weaken boundary sharpness between healthy and diseased regions. Similarly, for local variance computation, the window size n determines the spatial scale at which texture variability is measured. A small window captures fine-grained texture variations and sharp intensity transitions, which are important for detecting subtle disease patterns. In contrast, a larger window emphasizes broader texture distributions but may blur local discontinuities and reduce sensitivity to small infected areas. Therefore, the choice of n represents a trade-off between robustness to noise and preservation of spatial details. An intermediate window size provides a balanced representation, maintaining sufficient smoothness while preserving critical boundary information, which theoretically supports improved segmentation accuracy in visually ambiguous cactus disease images.

3.1.4 Model Training: Machine Learning Classifiers

After the image preprocessing and feature extraction stages, the resulting feature vectors were used to train and evaluate several machine learning classifiers, namely Random Forest (RF), Artificial Neural

Network (ANN), Support Vector Machine (SVM), and k-Nearest Neighbors (KNN). The extracted features include color descriptors (RGB intensity values), texture-based statistical measures derived from edge and gradient filters (Roberts, Sobel, Scharr, and Prewitt), as well as smoothing-based features obtained using Gaussian and median filters. These features collectively describe the local and global texture characteristics of healthy and diseased cactus regions. The dataset was divided into training and testing subsets, with the training set used for model learning and the testing set reserved for performance evaluation. To ensure fair comparison and reproducibility, the same feature set and data split strategy were applied to all classifiers.

Random Forest (RF): The Random Forest classifier was trained using an ensemble of decision trees constructed through bootstrap sampling. For each tree, a random subset of training samples and a random subset of features were selected at each split node. The final prediction was obtained by majority voting across all trees. This ensemble strategy improves robustness to noise and reduces overfitting, particularly in scenarios with limited data.

Artificial Neural Network (ANN): The ANN model consists of an input layer corresponding to the extracted features, one hidden layer with non-linear activation functions, and an output layer representing the disease classes. The network parameters were optimized using backpropagation to minimize the classification error. ANN was selected for its ability to model non-linear relationships among features; however, its performance is influenced by the limited dataset size.

Support Vector Machine (SVM): The SVM classifier was trained to determine an optimal separating hyperplane that maximizes the margin between different classes in the feature space. A radial basis function (RBF) kernel was employed to handle non-linear class boundaries. Despite its effectiveness in high-dimensional spaces, SVM performance was constrained by the strong visual similarity between healthy and diseased regions and the reliance on handcrafted features.

k-Nearest Neighbors (KNN): The KNN classifier assigns class labels based on the majority vote of the k nearest samples in the feature space, using the Euclidean distance metric. While KNN performs well on small datasets, its sensitivity to noise and the choice of k can affect classification accuracy.

3.2 Deep Learning Approach

3.2.1 Image pre-processing

For this work, 20 images and 20 masks of corresponding cacti, of original size 720 x 720, were

segmented into small pieces or "patches" of 256×256 pixels with a pitch of 200 pixels. This generated a total of 77 patches for images and masks. This method allows for smaller, easier-to-process image pieces for machine learning models. To compensate for the intensity of the sun, the brightness of the photos has been reduced. This step is crucial to ensure that segmented images contain visual information that is consistent and actionable by machine learning models. The PhotoRoom application was used for image preprocessing. This tool is popular for tasks such as background removal, which allows you to focus on the relevant areas of cactus images. Image preprocessing is an essential step to improve data quality and optimize the performance of machine learning models.

3.2.2 U-net Model

U-Net is a pretrained network model comprising an encoder, a bottleneck convolution layer, and a decoder, [31]. The encoder network, based on ResNet-34, consists of 34 convolutional blocks that encode the input images into feature representations at various levels with feature maps. These feature maps are progressively enhanced through each convolution block to effectively learn the complex patterns of cactus diseases. The input to the network is a patch $x \in R^{256 \times 256 \times 3}$, and the output is a segmentation map $\Phi(x) \in R^{256 \times 256 \times 4}$. The bottleneck convolution block transforms the features learned from the encoder network into the decoder networks. The decoder networks then semantically transform these discriminative features into pixel space to produce a dense output. Figure 2 (Appendix) illustrates the proposed U-Net architecture applied for cactus disease classification in this study. At the bottleneck stage, the ResNet-34 encoder outputs feature maps of size $16 \times 16 \times 512$ for an input patch of $256 \times 256 \times 3$, confirming the correct dimensional integration of the encoder-decoder architecture.

3.2.3 Residual Neural Network

Increasing the number of layers in a neural network can lead to a dense network, which may cause the vanishing gradient problem. As the network deepens, the weight updates during training can become ineffective, leading to challenges in learning and convergence. To address this issue, Residual Networks (ResNets) introduce skip connections that allow the network to bypass one or more layers. These skip connections help maintain effective gradient flow and mitigate the vanishing gradient problem.

In this study, ResNet-34 was utilized as the encoder network within the U-Net architecture.

ResNet-34's design facilitates efficient feature propagation by leveraging residual connections, which improves gradient flow and helps overcome the vanishing gradient challenges. This integration allows the U-Net model to better capture and analyze the complex features associated with cactus diseases, enhancing the overall performance of the disease detection system.

3.2.4 Encoder Network of U-Net

The encoder network of the U-Net architecture uses ResNet-34, a convolutional neural network (CNN) that is 34 layers deep and consists of 16 residual blocks. Each residual block performs downsampling of the input images, which helps in extracting high-level features from the data. This downsampling process is an integral part of the contracting path in U-Net, which progressively reduces the spatial dimensions while increasing the depth of feature maps. This enables the network to capture more complex and abstract features as it progresses deeper into the network. The encoder contains an initial block with a 7×7 convolution with 64 filters and a stride of 2, which decreases the input resolution by half. This allowed effectively enlarging the patches' size (256×256) to incorporate a larger context without a significant increase in computation and memory consumption. Each block doubles the number of feature channels and halves the spatial resolution. It uses two 3×3 convolutions, where the first one uses a stride of 2.

3.2.5 Decoder Network of the U Net

The decoder network is responsible for the expansive path of the U-Net architecture. It employs a series of transposed convolutions to up-sample the down-sampled features, effectively expanding the convolution size and reconstructing the spatial dimensions of the input images. The decoder network also integrates residual connections to maintain the feature information from the contracting path, enhancing the accuracy of the lesion boundary predictions and improving spatial resolution. The process begins with bilinear upsampling, followed by concatenation with the output from the corresponding encoder block via skip connections. This is followed by the application of two 3×3 convolutions with the same number of filters. Unlike the original decoder, we have added an extra block that does not receive any skip connection. The output of the final decoder block is then passed through a 1×1 convolution to generate the final score map. Additionally, each block in the decoder includes an extra 1×1 convolution, serving as a linear combination of the block's final feature maps to create an intermediate score map. These intermediate maps are utilized in two ways:

(a) for deep supervision and (b) to merge them through a linear combination for producing the final output. Apart from the 1×1 convolutions, a batch normalization layer and a ReLU activation function are applied immediately after each convolution throughout the network. Furthermore, appropriate padding is used to maintain the same spatial size during the convolutions.

4 Experiments and results

In this section, we detail the experiments conducted to evaluate the effectiveness of our proposed machine learning (ML) and deep learning (DL) approaches for the segmentation of cactus diseases. We describe the dataset, experimental configuration, evaluation metrics, and present the results obtained from our experiments. The segmentation model is designed to identify two components: cochineal (white) and black bacterial soft rot (black). Figure 3 (Appendix) and Figure 4 (Appendix) shows an example of visual segmentation outputs from our system. In this case, our approach successfully segments areas containing diseased cacti. The system accurately identifies the affected regions, demonstrating its effectiveness in real-world applications for disease detection and segmentation in cacti. The performance of the trained segmentation methods is evaluated on the test dataset. Test pixels are selected from representative regions of components to form a complete test dataset. This experiment of state-of-the-art machine learning and deep learning models was developed using the same heart failure data. All computations were performed on an Intel(R) Core(TM) i5-2520M CPU @ 2.50GHz 2.50 GHz, with 4.00 GB (3.88 GB usable) of RAM. The system used a 64-bit operating system, x64-based processor, running Windows 10. All the experiments were performed using the Python software package. Although the experimental setup was conducted on a relatively modest hardware configuration (Intel i5 CPU with 4.00 GB RAM), this environment was sufficient for the scope of the proposed experiments. The training process was performed using a patch-based strategy, where the original images were divided into smaller patches of 256×256 pixels. This approach significantly reduces memory requirements and computational load during both feature extraction and model training. Furthermore, the U-Net architecture was employed with a ResNet-34 encoder initialized using pretrained weights, which accelerates convergence and limits the need for extensive computational resources. The moderate batch sizes and controlled network depth ensured that memory consumption remained within available limits, preventing memory bottlenecks or numerical instabilities during training.

This research applied machine learning and deep

learning algorithms to segment and classify diseased cactus images. The goal was to compare models using accuracy, F1 score, and recall, which measure how well they detect and classify disease types. The performance of the proposed segmentation models was evaluated using standard pixel-level metrics, including Accuracy, Recall, Precision, and F1-score. Accuracy measures the proportion of correctly classified pixels and is defined as

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}, \quad (5)$$

where TP and TN denote the numbers of true positive and true negative pixels, while FP and FN represent false positive and false negative pixels, respectively. Recall, also known as sensitivity, evaluates the model's ability to correctly detect diseased regions and is expressed as

$$\text{Recall} = \frac{TP}{TP + FN}. \quad (6)$$

Precision reflects the reliability of pixels predicted as diseased and is given by

$$\text{Precision} = \frac{TP}{TP + FP}. \quad (7)$$

The F1-score provides a balanced evaluation by computing the harmonic mean of Precision and Recall, defined as

$$\text{F1-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}. \quad (8)$$

The results in Table 2 (Appendix) provide a comparative analysis of the performance of machine learning (ML) and deep learning (DL) models on a dataset containing images of cactus diseases, specifically Cochineal and Black Bacterial Soft Rot. The models were evaluated using accuracy, F1 score, and recall as performance metrics. This comparison of results can also be shown in graphical form, as illustrated in Figure 2.

The U-Net model showed the best results, with 0.98 accuracy, F1 score, and recall for Cochineal, and 0.94 for Black Bacterial Soft Rot. This confirms its strength in accurately segmenting disease regions in cactus images. The Random Forest (RF) model performed well, especially for Cochineal with 0.92 accuracy and F1 score, and 0.91 for Black Bacterial Soft Rot, ranking second after U-Net. SVM and KNN showed stable but lower results: SVM reached 0.87 for both diseases, while KNN scored 0.91 for Cochineal and 0.88 for Black Bacterial Soft Rot. The ANN achieved balanced results of 0.90 for both diseases, proving effective but less

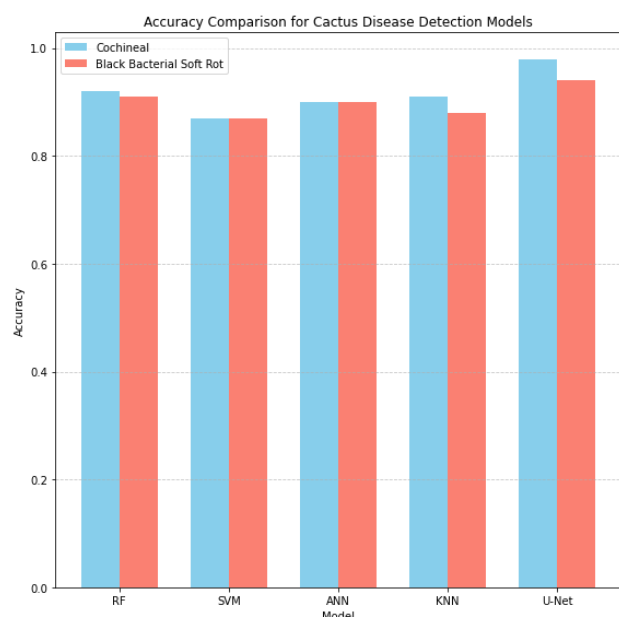


Fig. 2: Accuracy Comparison for Cactus Disease Segmentation Models.

Source: Created by the authors.

specialized for complex cases, the U-Net performance shown in Figure 3 highlights key learning patterns. Training accuracy rises sharply and stabilizes around epoch 50, while validation accuracy follows a similar trend but remains lower, suggesting slight overfitting. Training loss decreases consistently, reflecting effective learning, whereas validation loss stabilizes at a higher level and shows a slight late increase, further indicating mild overfitting.

For Cochineal, Figure 4 shows that U-Net quickly improved accuracy in the first 20 epochs and stabilized around 0.98 after 40 epochs for both training and validation, indicating strong learning and good generalization. The loss curves also decreased steadily, with validation loss stabilizing slightly above training loss after epoch 60, showing effective error minimization without significant overfitting. There are minor fluctuations in validation loss after epoch 100, which could indicate some sensitivity to variations in the validation data, but this is within an acceptable range. In this study, we compared the performance of different machine learning and deep learning models, including Random Forest (RF), Support Vector Machine (SVM), Artificial Neural Network (ANN), k-Nearest Neighbors (KNN), and U-Net, for the segmentation and detection of two cactus diseases: cochineal and black bacterial soft rot.

The slight increase in validation loss observed during training is primarily due to the limited dataset

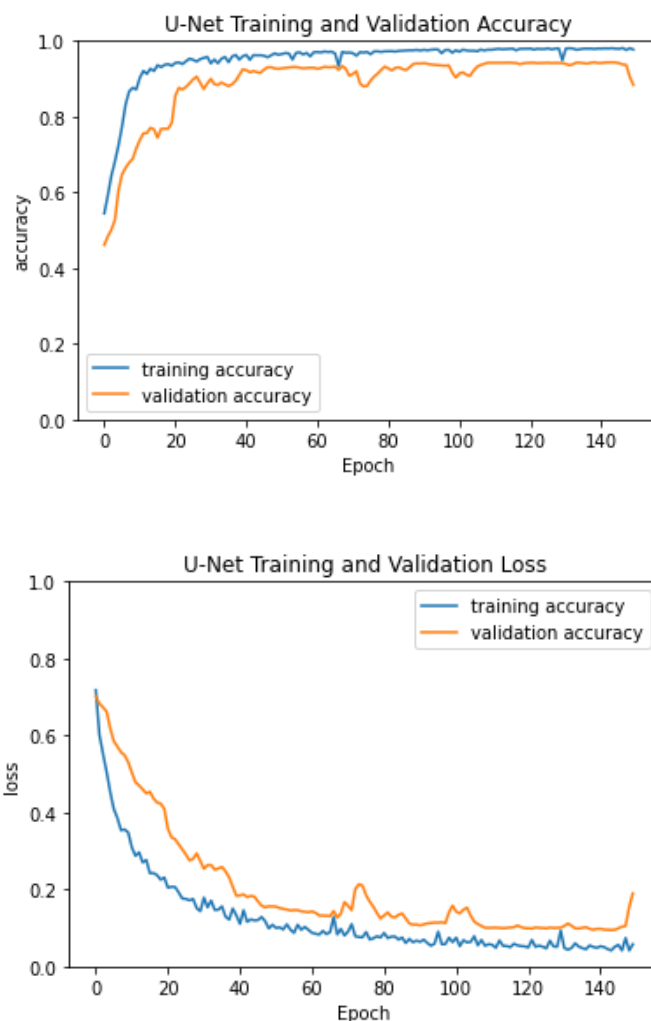


Fig. 3: Performance of U-net model with black soft.

Source: Created by the authors.

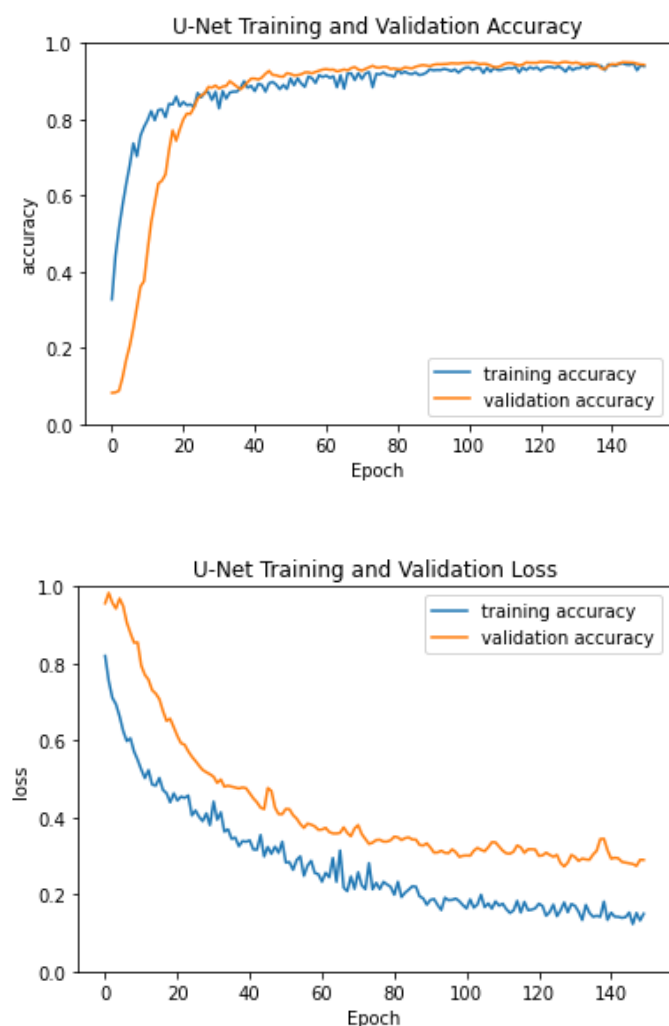


Fig. 4: Performance of U-net model with cochineal.

Source: Created by the authors.

size and the patch-based preprocessing strategy. 20 images and 20 masks of corresponding cacti, of original size 720 x 720, were segmented into small pieces or "patches" of 256 x 256 pixels. This generated a total of 77 patches for images and masks. This can cause the model to partially memorize repeated local patterns, resulting in mild overfitting. Despite this, numerical stability was maintained, as evidenced by the consistent convergence of training and validation losses and stable segmentation accuracy across epochs. The moderate patch size and stride also ensured memory efficiency, preventing computational bottlenecks. Overall, this mild overfitting is expected when training high-capacity models like U-Net on limited datasets and does not compromise the reliability of the results. The experimental results indicate that the SVM model consistently achieved the lowest accuracy among the evaluated approaches for both cochineal disease and black bacterial soft rot. This limitation can be primarily attributed to the high visual similarity between healthy and diseased cactus regions, which results in overlapping feature distributions when handcrafted texture descriptors are employed. Consequently, the discriminative capability of SVM becomes constrained in scenarios where subtle visual differences dominate the image content. Moreover, the handcrafted features used by the SVM model are inherently limited in their ability to represent complex non-linear and spatial relationships that characterize plant disease patterns, particularly during early infection stages where lesion boundaries are poorly defined. As a result, the model struggles to effectively separate visually similar regions, leading to reduced classification and segmentation performance. In contrast, the Random Forest model benefits from its ensemble-based architecture, which enhances robustness to noise and feature variability by aggregating multiple decision trees. This property enables improved generalization, especially under limited data conditions. The U-Net model significantly outperforms all classical machine learning approaches by learning hierarchical and contextual representations directly at the pixel level. Its encoder-decoder structure, combined with skip connections, allows for the integration of fine-grained spatial details with high-level semantic information, thereby enabling more accurate delineation of subtle disease regions in visually ambiguous scenarios. designed to keep both fine details and overall structure using its encoder-decoder with skip connections. While the encoder reduces image size to capture high-level features, some boundary details could be lost. The skip connections fix this by sending these details directly to the decoder, helping it combine global context with local information.

This explains why U-Net achieves high accuracy for both cochineal and black bacterial soft rot, especially when healthy and diseased cactus areas look very similar. The results confirm that U-Net effectively preserves edges and improves boundary detection in complex images.

5 Conclusion and Future Works

For cochineal disease, the U-Net model achieved the highest performance across all metrics, with an accuracy, F1 score, and recall of 0.98, significantly outperforming other models. This underscores U-Net's robustness and precision in accurately identifying disease regions in cactus images. Random Forest also showed strong results with an accuracy and F1 score of 0.92, making it a viable option for disease detection when deep learning resources are limited. KNN and ANN also provided balanced results, with accuracies of 0.91 and 0.90, respectively, while SVM achieved slightly lower performance with an accuracy of 0.87.

For black bacterial soft rot, U-Net maintained its high performance, achieving an accuracy, F1 score, and recall of 0.94, indicating its adaptability across different disease types. Random Forest remained the second most effective model for this disease, with an accuracy of 0.91. Both ANN and KNN showed stable but slightly lower accuracies of 0.90 and 0.88, respectively, while SVM consistently provided an accuracy of 0.87.

The results suggest that U-Net, with its ability to capture complex spatial relationships, is the most suitable model for cactus disease segmentation. However, Random Forest also proves to be effective for disease classification, particularly for scenarios with limited computational resources or simpler classification requirements.

For future research, we plan to investigate other deep learning architectures like LinkNet to further enhance segmentation accuracy and computational efficiency. Additionally, exploring hybrid approaches that combine the strengths of both traditional machine learning (e.g., Random Forest) and deep learning (e.g., U-Net) could lead to more robust and adaptable solutions for large-scale agricultural disease monitoring.

References:

- [1] UshaDevi, G., and Gokulnath, B. V., "A Survey on Plant Disease Prediction Using Machine Learning and Deep Learning Techniques," *Inteligencia Artificial*, vol. 23, no. 65, pp. 136–154, 2020. <https://doi.org/10.4114/intartif.vol23iss65pp136-154>
- [2] Ramanjot, Mittal, U., Wadhawan, A., Singla, J., "Plant Disease Detection and Classification: A Systematic Literature Review," *Sensors*, vol. 23, no. 10, p. 4769, 2023. <https://doi.org/10.3390/s23104769>
- [3] Liu, J., and Wang, X., "Plant Diseases and Pests Detection Based on Deep Learning: A Review," *Plant Methods*, vol. 17, no. 22, 2021. <https://doi.org/10.1186/s13007-021-00722-9>
- [4] Lamba, M., Gigras, Y. and Dhull, A., "Classification of Plant Diseases Using Machine and Deep Learning," *Open Computer Science*, vol. 11, no. 1, pp. 491–508, 2021. <https://doi.org/10.1515/comp-2020-0122>
- [5] Demilie, W. B., "Plant disease detection and classification techniques: A comparative study of the performances," *Journal of Big Data*, vol. 11, article no. 5, 2024. <https://doi.org/10.1186/s40537-023-00863-9>
- [6] Zhang, N., Yang, G., Pan, Y., Yang, X., Chen, L. and Zhao, C., "A Review of Advanced Technologies and Development for Hyperspectral-Based Plant Disease Detection in the Past Three Decades," *Remote Sensing*, vol. 12, no. 19, 3188, 2020. <https://doi.org/10.3390/rs12193188>
- [7] Rao, D. S., Babu Ch, R., Kiran, V. S., Rajasekhar, N., Srinivas, K., Akshay, P. S., Mohan, G. S., and Bharadwaj, B. L., "Plant Disease Classification Using Deep Bilinear CNN," *Intelligent Automation und Soft Computing*, vol. 31, no. 1, pp. 161–176, 2021. <https://doi.org/10.32604/iasc.2022.017706>
- [8] L. Mohammed and Y. Yusoff, "Detection and Classification of Plant Leaf Diseases Using Digital Image Processing Methods: A Review," *ASEAN Engineering Journal*, vol. 13, no. 1, pp. 1–9, 2023. <https://doi.org/10.11113/aej.v13.17460>
- [9] Priya, P. and Dsouza, D. A., "Study of Feature Extraction Techniques for the Detection of Diseases of Agricultural Products," *International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering*, vol. 3, no. 5, pp. 234–239, 2015. https://ijireeice.com/wp-content/uploads/2015/05/T1_102.pdf
- [10] Sannakki, S. S., Rajpurohit, V. S., Nargund, V. B., and Kulkarni, P., "Diagnosis and

Classification of Grape Leaf Diseases using Neural Network,” Proceedings of the IEEE International Conference on Communication, Computing and Networking Technologies (ICCCNT), Tiruchengode, India, 2013, pp. 1–5. <https://doi.org/10.1109/ICCCNT.2013.6726616>

- [11] Arora, P., and Vikas, O., ”Relation Inclusive Search for Hindi Documents,” International Journal of Advanced Computer Science and Applications, vol. 4, no. 7, 2013. <https://doi.org/10.14569/IJACSA.2013.040701>
- [12] Jalal, A. S. and Dubey, S. R., ”Detection and Classification of Apple Fruit Diseases Using Complete Local Binary Patterns,” Proceedings of the IEEE Third International Conference on Computer and Communication Technology (ICCCCT), Allahabad, India, 2012, pp. 1–6. <https://doi.org/10.1109/ICCCCT.2012.76>.
- [13] Mokhtar, U., Ali, M. A. S., Hassenian, A. E., and Hefny, H., ”Tomato Leaves Diseases Detection Approach Based on Support Vector Machines,” Proceedings of the IEEE International Conference on Computer Engineering (ICENCO), Cairo, Egypt, 2015, pp. 246–250. <https://doi.org/10.1109/ICENCO.2015.7416356>
- [14] Khirade, S. D., and Patil, A. B., ”Plant Disease Detection Using Image Processing,” in *Proceedings of the IEEE International Conference on Computing Communication Control and Automation (ICCUBEA)*, Pune, India, 2015, pp. 768–771. <https://doi.org/10.1109/ICCUBEA.2015.153>
- [15] Bhimte, N. R., and Thool, V. R., ”Diseases Detection of Cotton Leaf Spot Using Image Processing and SVM Classifier,” in *2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS)*, Madurai, India, 2018, pp. 340–344. <https://doi.org/10.1109/ICCONS.2018.8662906>
- [16] Kaur, L., and Sharma, S. G., ”Identification of Plant Diseases and Distinct Approaches for Their Management,” Bulletin of the National Research Centre, vol. 45, no. 169, 2021. <https://doi.org/10.1186/s42269-021-00627-6>
- [17] Patil, R., Deshpande, V. S., Sable, N. P., and Mahalle, P. N., ”Ensemble CNN-Based Cotton Disease Detection and Pest Recommendation System for Indian Agriculture,” Proceedings of the IEEE International Conference on Recent Advances in Computing and Communication Systems, India, 2023, pp. 1–6. <https://doi.org/10.1109/ICICIS65613.2025.11371076>
- [18] Lu, Y., Yi, S., Zeng, N., Liu, Y., and Zhang, Y., ”Identification of rice diseases using deep convolutional neural networks,” Neurocomputing, vol. 267, pp. 378–384, 2017. <https://doi.org/10.1016/j.neucom.2017.06.023>
- [19] Prajwala, T. M., Sharma, A., and Reddy, K., ”Tomato Leaf Disease Detection Using a Modified Convolutional Neural Network Model,” Proceedings of the International Conference on Smart Technology and Applications, Bangalore, India, 2020, pp. 50–55. <https://doi.org/10.1109/IC3.2018.8530532>
- [20] Biswas, S., Kumar, M., and Verma, R., ”Severity Determination of Potato Disease Using Image Processing and Neural Networks,” Proceedings of the IEEE International Humanitarian Technology Conference (IHTC), Montreal, Canada, 2014, pp. 1–5. <https://doi.org/10.1109/IHTC.2014.7147519>
- [21] Kurmi, Y., Gangwar, S., Agrawal, D., Kumar, S., and Srivastava, H. S., ”Leaf image analysis-based crop diseases classification,” Signal, Image and Video Processing, vol. 15, no. 3, pp. 589–597, 2021. <https://doi.org/10.1007/s11760-020-01780-7>
- [22] Jaisakthi, S. M., Mirunalini, P., Thenmozhi, D., and Vatsala, ”Grape Leaf Disease Identification using Machine Learning Techniques,” Proceedings of the International Conference on Computational Intelligence in Data Science (ICCIDS), India, 2019, pp. 1–6. <https://doi.org/10.1109/ICCIDS.2019.8862084>
- [23] Piska, G., Janaskar, S., Chandgadkar, O., Bhawsar, P., and Vishwakarma, P. P., ”Enhancing plant disease detection using machine learning approaches for improved agricultural productivity,” Journal of Computer and Agricultural Applications, vol. 15, no. 2, 2020. <https://doi.org/10.11591/ijece.v15i2.pp2081-2088>
- [24] Aqel, D., Al-Zubi, S., Mughaid, A., and Jararweh, Y., ”Extreme learning machine for plant diseases classification: a sustainable approach for smart agriculture,” Cluster Computing, vol. 25, p. 2303, 2022. <https://doi.org/10.1007/s10586-021-03397-y>

- [25] M. Sokolova and G. Lapalme, "A Systematic Analysis of Performance Measures for Classification Tasks," *Information Processing and Management*, Vol. 45, No. 4, 2009, pp. 427–437. <https://doi.org/10.1016/j.ipm.2009.03.002>
- [26] Wang, S., Xu, D., Liang, H., Bai, Y., Li, X., Zhou, J., Su, C., and Wei, W., "Advances in Deep Learning Applications for Plant Disease and Pest Detection: A Review," *Remote Sensing*, vol. 17, no. 4, p. 698, 2025. <https://doi.org/10.3390/rs17040698>
- [27] Too, E. C., Yujian, L., Njuki, S., and Yingchun, L., "A comparative study of fine-tuning deep learning models for plant disease identification," *Computers and Electronics in Agriculture*, vol. 161, pp. 272–279, 2019. <https://doi.org/10.1016/j.compag.2018.03.032>
- [28] Hoang, N.-D. and Nguyen, Q.-L., "Metaheuristic Optimized Edge Detection for Recognition of Concrete Wall Cracks: A Comparative Study on the Performances of Roberts, Prewitt, Canny, and Sobel Algorithms," *Advances in Civil Engineering*, vol. 2018, Article ID 7163580, 16 pages, 2018. <https://doi.org/10.1155/2018/7163580>
- [29] Goyal, B., Dogra, A., Agrawal, S., Sohi, B. S. and Sharma, A., "Image Denoising Review: From Classical to State-of-the-Art Approaches," *Information Fusion*, vol. 55, pp. 220–244, 2020. <https://doi.org/10.1016/j.inffus.2019.09.003>
- [30] Hesamian, M. H., Jia, W., He, X., and Kennedy, P., "Deep Learning Techniques for Medical Image Segmentation: Achievements and Challenges," *Journal*

of Imaging Informatics in Medicine, vol. 32, no. 5, pp. 582–596, 2019. <https://doi.org/10.1007/s10278-019-00227-x>

- [31] Yadav, R. and Seth, A., "Deep Learning Based Approach for Plant Disease Detection," Preprint (ResearchSquare), 2024. <https://doi.org/10.21203/rs.3.rs-2170464/v1>

Contribution of Individual Authors to the Creation of a Scientific Article (Ghostwriting Policy)

Aissam El Ibrahimi conceptualized and formulated the research problem, performed the experiments, collected and analyzed the data, and wrote the original draft of the manuscript. Nabil El Akchioui supervised the research, provided methodological guidance, reviewed the manuscript, and contributed to the discussion of results. Both authors to the present research at all stages, from problem formulation to final findings and solution.

Sources of Funding for Research Presented in a Scientific Article or Scientific Article Itself

No funding was received for conducting this study.

Conflict of Interest

The authors have no conflicts of interest to declare that are relevant to the content of this article.

Creative Commons Attribution License 4.0 (Attribution 4.0 International, CC BY 4.0)

This article is published under the terms of the Creative Commons Attribution License 4.0 International (CC BY 4.0), which permits use, distribution, and reproduction in any medium, provided the original work is properly cited. For more details, see <https://creativecommons.org/licenses/by/4.0/>.

APPENDIX

Table 1. Features Used for Random Forest Algorithm

Index	R	G	B	Roberts	Sobel	Scharr	Prewitt	Gaussian	Median	Labels
1	100	110	120	0.02511	0.012555	0.012942	0.012052	117	119	76
2	98	108	118	0.039216	0.01308	0.017541	0.002219	117	119	76
3	105	115	125	0.011433	0.011927	0.011059	0.013137	118	119	76
4	5	10	15	0.024802	0.009301	0.009562	0.010209	14	16	29
5	2	7	12	0.020188	0.007843	0.009327	0.005918	15	15	29
6	2	7	12	0.016857	0.00932	0.007696	0.007696	15	15	29
7	203	208	213	0.123669	0.238324	0.240302	0.235911	176	213	226
8	224	229	234	0.068487	0.183113	0.183384	0.183966	188	234	226
9	232	237	242	0.05098	0.19426	0.195104	0.19139	197	242	226
10	71	76	81	0.072987	0.07845	0.07804	0.07909	81	81	150
11	71	76	81	0.063869	0.036165	0.036145	0.033188	80	81	150
12	74	79	84	0.047222	0.05176	0.050194	0.051894	81	82	150

Source: Created by the authors.

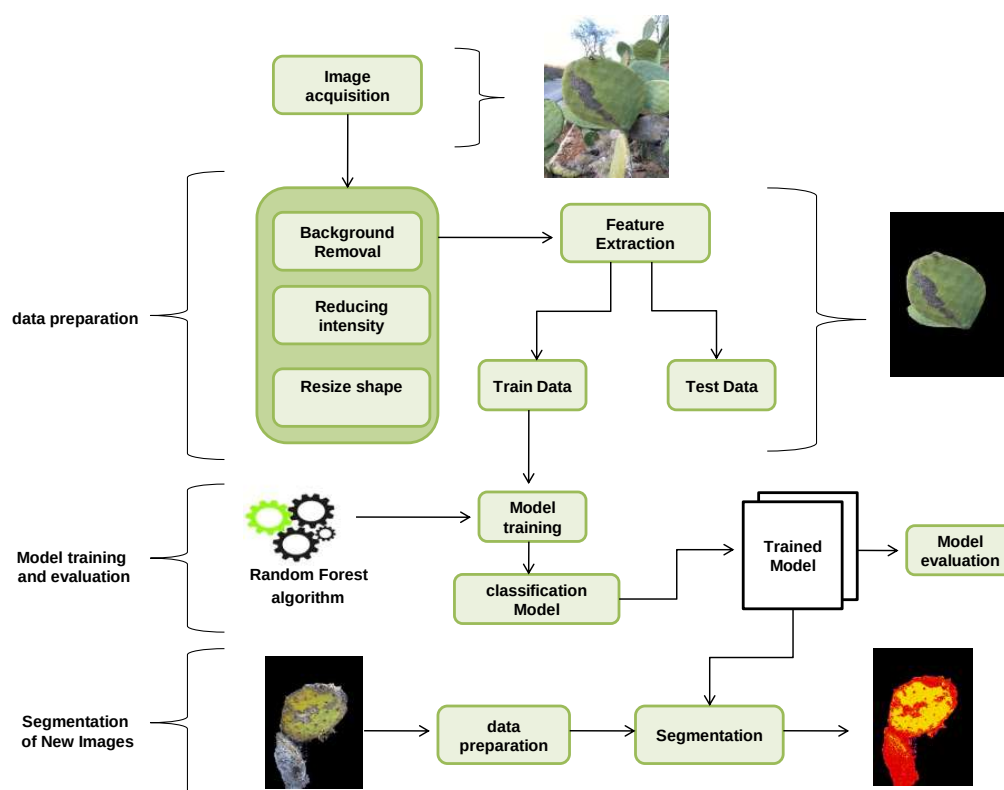


Fig. 1: Proposed methodology for Cactus disease Segmentation.

Source: Created by the authors.

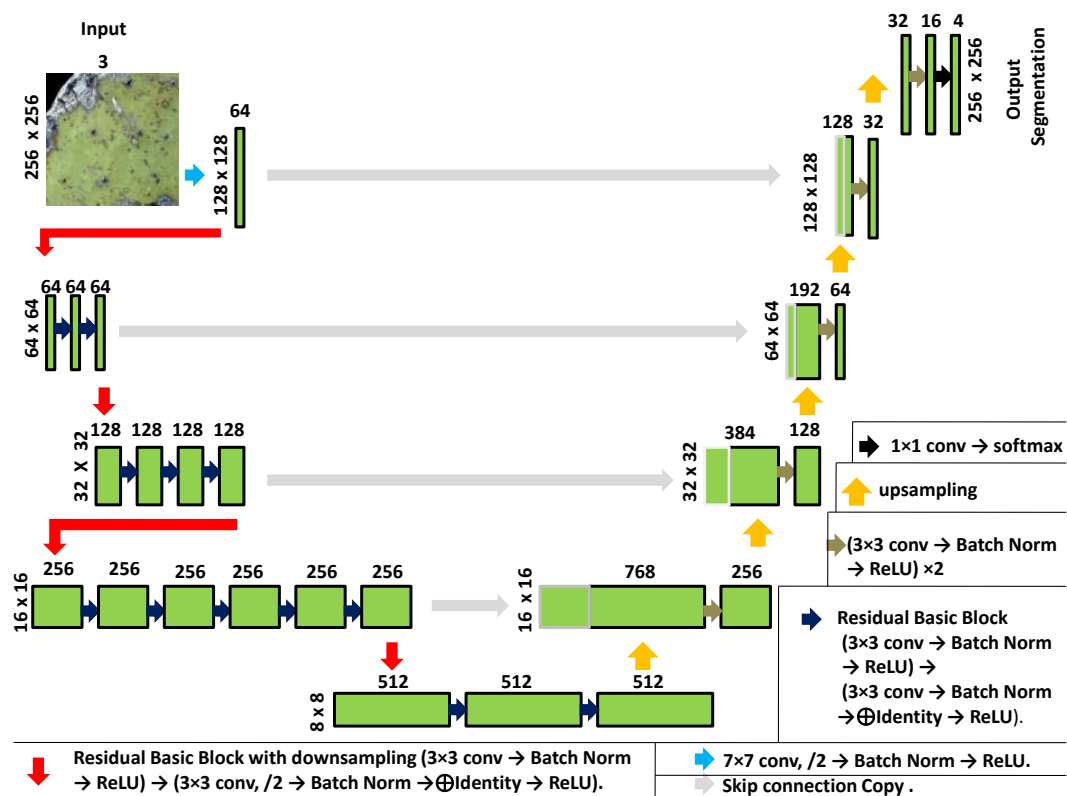


Fig. 2: U-Net architecture with a ResNet-34 encoder.
Source: Created by the authors.

Table 2. Performance Comparison on Cactus Disease Segmentation Models.

2*Type of Patient	2*Model	Performance Metrics		
		Accuracy	F1 Score	Recall
5*Cochineal	RF	0.92	0.92	0.92
	SVM	0.87	0.85	0.87
	ANN	0.90	0.90	0.90
	KNN	0.91	0.91	0.91
	U-Net	0.98	0.98	0.98
5*Black Bacterial Soft Rot	RF	0.91	0.91	0.91
	SVM	0.87	0.87	0.87
	ANN	0.90	0.90	0.90
	KNN	0.88	0.88	0.88
	U-Net	0.94	0.94	0.94

Source: Created by the authors.

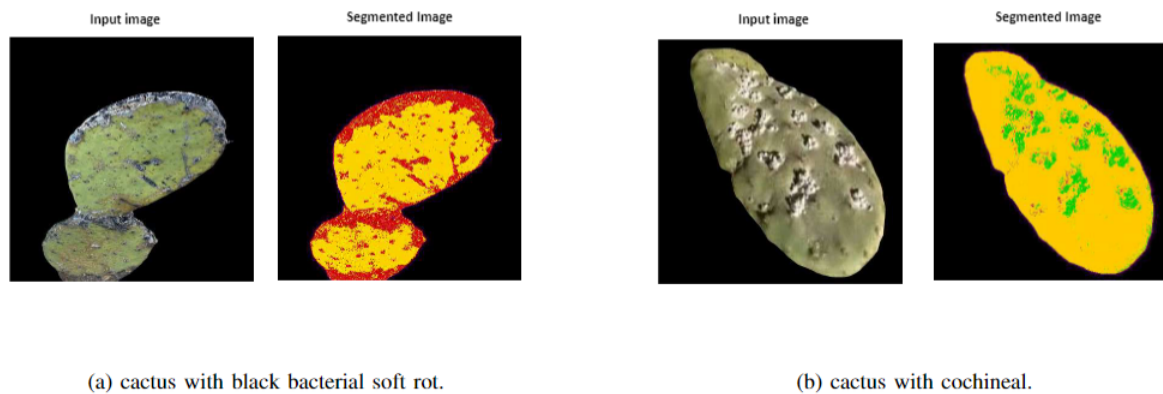


Fig. 3: Example of visual segmentation, obtained using our Machine Learning Approach with RF.
Source: Created by the authors.

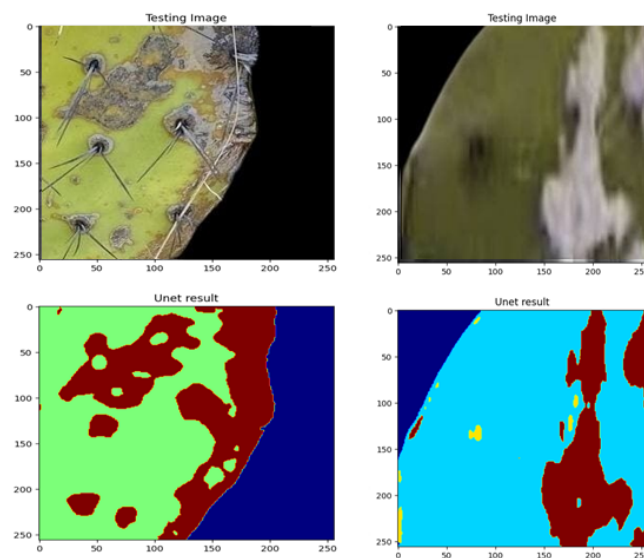


Fig. 4: Example of visual segmentation, obtained using our Deep Learning Approach with U net.
Source: Created by the authors.