Nanotechnology Perceptions
ISSN 1660-6795
www.nano-ntp.com

Enhancing Disease Classification Accuracy in Tomato Plants through Watershed Segmentation and Machine Learning Techniques

Murali B[1], Nagaraju C[2]

[1]Research Scholar, Department of Computer Science and Engineering, YSR Engineering College of YVU, India, bmuralicse@gmail.com
[2]Professor, Department of Computer Science and Engineering, YSR Engineering College of YVU, India.

Agriculture plays a crucial role in the global economy and food supply, with tomatoes being particularly important due to their nutritional and culinary value. Effective management of tomato plant diseases is essential for maintaining productivity and ensuring a stable food supply. Predominant diseases affecting tomato leaves, such as Bacterial Spot, Tomato Mosaic Virus, and Tomato Yellow Leaf Curl Virus, pose significant challenges to crop yield. Addressing these concerns, our study presents a refined machine learning-based methodology to improve the accuracy of disease classification in tomato plants. Our study, the methodology involves a series of steps including pre-processing, morphological reconstruction, and the formulation of a multi-scale gradient image. This process incorporates the use of both internal and external marker images to accomplish precise segmentation, followed by the generation of a watershed segmented image. This crucial step facilitates the extraction of distinctive features that are vital for the ensuing classification stage. Our evaluation contrasts various machine learning classifiers, namely Support Vector Machines (SVM), Random Forest Classifier (RFC), Logistic Regression, and XGBoost, within the context of this advanced methodology. In the existing method, SVM registered an accuracy of 91%, while Random Forest, Logistic Regression, and XGBoost each attained an accuracy of 90%. Enhancement of the method by including hue moments, haralick features, and the novel addition of Gabor features, culminated in a significant uptick in performance. The proposed approach has achieved better results than the existing method, with the SVM classifier attaining a superior accuracy of 93%, compared to the existing SVM's accuracy of 91%.

Keywords: Support Vector Machines (SVM), Watershed Segmentation, Machine Learning, Tomato Plant Diseases, Feature Extraction.

1. Introduction

Agriculture plays a pivotal role in the global economy, providing food, raw materials, and employment to billions of people. Among the myriad of crops cultivated, the tomato plant

(Solanum lycopersicum) stands out due to its nutritional value and widespread consumption. However, the productivity and quality of tomato crops are severely threatened by various diseases, including Bacterial Spot, Tomato Mosaic Virus, and Tomato Yellow Leaf Curl Virus. These diseases not only diminish the yield but also affect the quality of the tomatoes, leading to significant economic losses worldwide. The causes of these diseases are multifaceted, involving complex interactions between the tomato plant, pathogens, and environmental conditions. Bacterial Spot is caused by different species of Xanthomonas, which are capable of infecting the plant through wounds or natural openings, leading to necrotic lesions. Tomato Mosaic Virus, a member of the Tobamovirus genus, is mechanically transmitted and can cause systemic necrosis in susceptible tomato cultivars. Tomato Yellow Leaf Curl Virus, a Begomovirus transmitted by whiteflies, induces severe symptoms such as yellowing and leaf curling, significantly reducing the plant's photosynthetic capacity (Abrahamian et al., (2021); Potnis et al., (2015); Czosnek and Laterrot (1997); (Xu et al. (2021); Navas- Castillo et al. (1999); Wang et al., (2022); Shen et al., (2020)). Geetha et al., (2020) conducted a study focusing on the recognition and classification of diseases in tomato plants, which are crucial for preventing significant losses in quantity and yield. They utilized image processing techniques and algorithms to address these issues. The study consisted of four stages: pre-processing, leaf segmentation, feature extraction, and classification. Pre-processing was used to remove noise, while image segmentation was employed to isolate affected areas of the leaves. The k-nearest neighbors (KNN) algorithm, a supervised machine learning approach, was implemented for classification and regression tasks. The study also recommended treatment based on the identified diseases. Overall, the paper presented a method for leaf disease detection using image processing, which provides quick and reliable results to farmers based on color, shape, and texture analysis. Zhao et al. (2008) discusses an improved watershed algorithm based on opening-closing operations and distance transform to overcome the over-segmentation issue common in classical watershed segmentation. Their method retains the advantages of the watershed algorithm based on distance transform, enabling successful segmentation of individual dowels in images, which aids in computer vision and automatic counting tasks. The technique also preserves the original edges of each dowel, overcoming the over-segmentation problem typically associated with traditional watershed segmentation. Li and Li (2014) introduce an enhanced watershed algorithm for bridge image segmentation to address the over-segmentation issue. They utilize top-hat and bottom-hat transformations for image filtering, a multiscale approach for computing morphological gradient images, and an automatic calculation of the marker-extraction threshold based on gradient map statistics. This improved algorithm incorporates region merging based on fisher distance and adheres to the divergence principle to resolve over-segmentation in initial watershed segmentation, demonstrating its feasibility and effectiveness through various experiments. Zhang and Xu (2012) presents an improved watershed algorithm based on gradient transform, open-close reconstruction, and distance transform, specifically designed to mitigate the sensitivity to noise of traditional watershed algorithms. Their experiments show that this enhanced algorithm performs well in cell image segmentation, indicating its potential for effective application in similar domains. Hai et al. (2016) proposes a medical image segmentation approach combining K-means clustering and an improved watershed algorithm. They introduce the concept of similarity to refine the watershed algorithm and merge adjacent tiles from the initial segmentation. Their method effectively addresses the over-segmentation

issue prevalent in traditional watershed techniques, achieving satisfactory results in magnetic resonance image segmentation. Nasir and Fajri. (2019) conducted a study focusing on the management of rice plants, which are a staple food in Indonesia. The study aimed to increase production and farmer opinion through the innovative and dynamic use of technology in collaboration with farmers. Specifically, the researchers aimed to develop appropriate technology facilities to help farmers identify leaf diseases that occur during the growing period of rice, thereby enhancing production. The study applied the Gray Level Co-Occurrence Matrix method to extract feature values from rice leaves and used differences in leaf patterns to identify each rice leaf pattern, leading to the diagnosis of leaf diseases. The accuracy of the system developed in this study was reported to be 72.5%. Xian and Ngadiran. (2021) conducted a study on plant disease classification using image analysis and machine learning techniques, specifically Extreme Learning Machine (ELM). The researchers pre-processed leaf images and extracted features using Haralick textures. They trained an ELM classifier on a dataset of tomato plant leaves and achieved an accuracy of 84.94% in disease classification, outperforming models like Support Vector Machine and Decision Tree. This research has implications for early disease detection in agriculture, potentially improving crop yield and food security. Jos and Venkatesh. (2020) conducted a study on color-based texture classification for identifying plant diseases using machine vision. They applied texture features from literature to assess their effectiveness for Mango and Tomato plants. The study evaluated different statistical functions and found that Gray Level Co-occurrence based statistics were effective for distinguishing between smooth new leaves, dry leaves, and growth patterns. However, these values failed to discriminate tomato diseases effectively. To address this, the researchers proposed a novel method utilizing second-order statistics on a pseudo-color based co-occurrence matrix, resulting in improved classification for three tomato diseases. The study suggests that this method could be applied for early disease detection in various plants, assisting farmers in taking corrective measures to prevent yield loss. Manik et al., (2020) conducted research on plant classification based on leaf texture using image analysis techniques. Indonesia's rich plant diversity necessitated strategic steps for recording and identifying plants in the country. The Gray Level Co-occurrence Matrix (GLCM) method was employed for feature extraction from leaf images. The k-Nearest Neighbor (k-NN) method was implemented for classification, with the research showing that the accuracy of classification using GLCM with a k value of 3 was 83%. It was observed that the choice of the parameter k influenced the classification results, with higher k values leading to lower accuracy. Some classification errors occurred due to the similarity in the extracted traits from GLCM, resulting in a narrow range of values. Jeyalakshmi and Radha. (2020) aimed to automatically classify diseased potato and grape leaves from healthy ones. They used a sample size of 3000 and 4270 images for potato and grape leaves, respectively, obtained from the PlantVillage dataset. Color features such as average Red, Green, Blue, and Hue intensities of the lesion region were calculated, along with texture features including Contrast, Dissimilarity, Homogeneity, Energy, Correlation, ASM, and Entropy from the lesion region. Additionally, histogram features such as mean and standard deviation were extracted from the infected region. Data normalization was performed to bring all features to a common scale. Naïve Bayes, K Nearest Neighbor, and Support Vector Machine classifiers were applied to the feature sets. The dataset was split into 80% for training and 20% for testing. The classifiers achieved accuracies of 88.67%, 94.00%, and 96.83% for potato leaves, and 81.87%, 93.10%,

and 96.02% for grape leaves, respectively. The SVM classifier exhibited the highest accuracy for both species. The proposed method demonstrated superior performance compared to related works in the literature, highlighting its effectiveness in classifying grape and potato diseases. Samrin. (2021), focused on classifying plant species images to distinguish between crop seedlings and weeds. The study addressed the challenge of background noise, specifically stones in the images, which could interfere with machine learning predictions. To mitigate this, color-based segmentation was applied to isolate the green color pixels corresponding to the seedlings, effectively removing the stones from the background. This resulted in clean plant images ready for model input. Weed control was highlighted as a significant challenge for farmers, leading to decreased plant productivity. The study proposed weed classification as a solution to improve plant productivity by effectively managing weed growth in farming. Arjunagi and Patil. (2019) explored the application of machine learning techniques in agricultural contexts, particularly in the identification and classification of leaf diseases. The study employed triangular-based and OTSU-based methods for segmentation, followed by the extraction of textural features using the Gray Level Co-Occurrence Matrix (GLCM) and k-means clustering technique. Subsequently, a Support Vector Machine (SVM) classifier was utilized for the classification of different leaf diseases. The proposed method achieved an overall classification accuracy of 70% using the triangular-based segmentation, with an Area Under the Curve (AUC) of 0.63. In the context of enhancing disease classification in tomato plants, algorithms play a crucial role in both preprocessing and feature extraction phases. These algorithms are designed to improve image quality and extract meaningful information that can be used for accurate classification. Below is a definition of the algorithms and their importance, followed by the specific steps for each algorithm with related references.

The existing method for tomato disease classification leverages a combination of image preprocessing techniques, including median filtering for noise reduction and morphological reconstruction to enhance the features of interest, followed by multi-scale gradient image generation. Feature extraction involved hue moments and Haralick features. Various machine learning algorithms, including Support Vector Machines (SVM), Random Forest Classifier (RFC), Logistic Regression, and XGBoost, were used for classification. The SVM classifier achieved an accuracy of 91%, while the other classifiers showed similar performance levels, around 90-91%. Although effective, this method had limitations, particularly in addressing over-segmentation issues during the image preprocessing phase, which affected the accuracy of disease detection.

2. MATERIALS AND METHODS

1. Dataset Selection and Preprocessing

For this study, a dataset comprising 9448 images sourced from the PlantVillage dataset was chosen. The dataset contained both healthy and diseased tomato leaf samples, categorised into four classes, including images of healthy tomato leaves as well as leaves affected by bacterial spot, tomato mosaic, and yellow leaf curl. To facilitate model assessment, the dataset was split into training (80%) and testing (20%) sets.

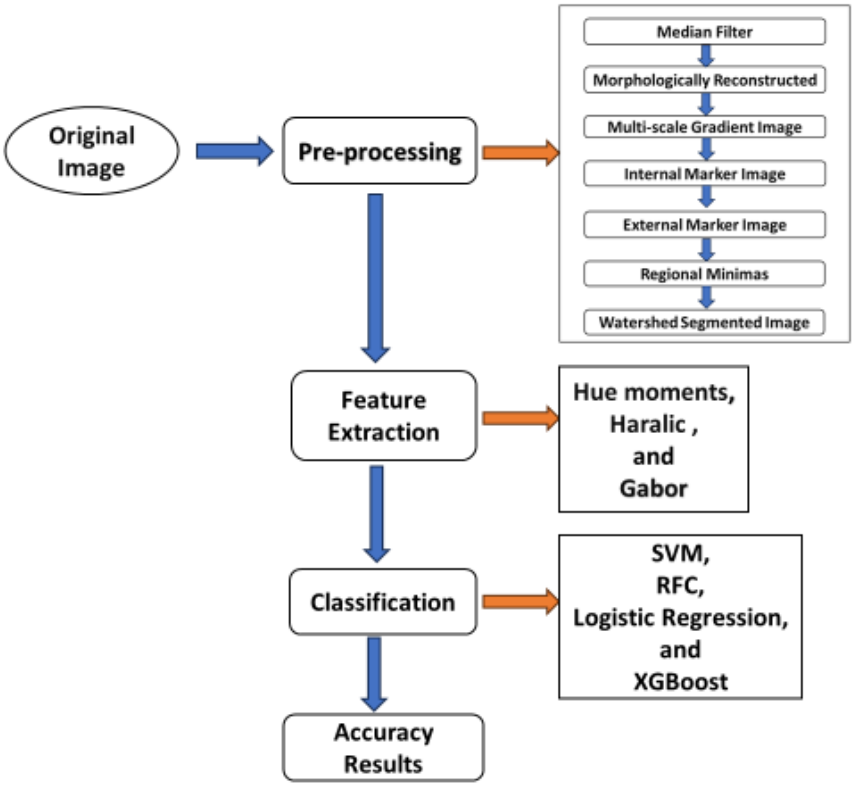


Fig.1. Proposed Methodology

1.1. Median Filter

The Median Filter is a non-linear digital filtering technique, often used in image processing, to reduce noise. It operates by moving a window (or kernel) across the image, replacing each entry with the median of neighboring entries. The kernel size, usually an odd number like 3x3, 5x5, etc., determines the number of neighbors considered in the calculation. This technique is particularly effective at preserving edges while removing noise, as it maintains the sharpness of sudden intensity changes. (Sebastiani and Stramaglia. (1997); Liang and Gao. (2013))

Given an image I and a pixel location (x,y), the median filter operation MF with a kernel size k can be defined as:

$$MF (I (x , y), k) = \text{median}\{ I (x + i, y + j) | i , j \in W\}$$

where W is the set of offsets defining the neighborhood around the pixel (x,y) within the kernel window, with $i,j \in [-\frac{k}{2}, \frac{k}{2}]$.

Algorithm

1. Pad the input image with zeros on all sides to handle border pixels.

- Padding size = $\frac{k-1}{2}$

2. For each pixel (i,j) in the padded image:

2.1. Extract a window of size kernel_size×kernel_size centered at (i,j).

2.2. Sort the pixel values within the window in ascending order.

2.3. Compute the median value from the sorted window:

- If kernel_size is odd, median = sorted_window $[\frac{k^2-1}{2}]$

- If kernel_size is even (less common), median =

$$\frac{\text{sorted\_window}[\frac{k^2}{2}-1]+\text{sorted\_window}[\frac{k^2}{2}]}{2}$$

2.4. Set the output pixel value at (i,j) in the filtered_image to the computed median.

3. Return the filtered_image.

Input:

- image: Input image to be filtered
- kernel_size: Size of the filtering window (an odd integer)

Output:

- filtered_image: Median filtered image

1.2. Morphological Reconstruction

Morphological reconstruction is a process in image processing that is used to extract relevant shapes and structures from images. It is particularly useful in applications such as plant disease detection, where it can help to identify and analyze affected areas on plant leaves or stems. The algorithm typically involves the use of morphological operations like dilation and erosion, which are defined by specific mathematical expressions (Vijaya Kumar et al., (2010)).

Morphological Operations

The two primary morphological operations are:

Dilation: It adds pixels to the boundaries of objects in an image, expanding them. The mathematical expression for dilation of an image f by a structuring element B is given by:

$$(f \oplus B)(x,y) = \max_{(s,t)\in B}\{f(x-s,y-t)\}$$

Erosion: It removes pixels on object boundaries, effectively shrinking the objects in an image. The mathematical expression for erosion of an image f by a structuring element B is:

$$(f \ominus B)(x,y) = \min_{(s,t)\in B}\{f(x+s, y+t)\}$$

Morphological Reconstruction Algorithm

The morphological reconstruction algorithm can be described as follows:

1. Marker Image: Start with a marker image f that is derived from the original image but is "lower" than the original image. This marker image highlights the features to be preserved or reconstructed.

2. Mask Image: The original image serves as the mask image g, which constrains the reconstruction.

3. Iteration: Apply the dilation to the marker image, and then intersect it with the mask image. Repeat this process iteratively:

$$f^{(k+1)} = (f^{(k)} \oplus B) \wedge g$$

where $f^{(k)}$ is the marker image at iteration k, B is the structuring element, and $\wedge$ represents the pointwise minimum with the mask image g.

4. Convergence: Continue the iteration until stability is reached, i.e.,

$$f^{(k+1)} = f^{(k)}.$$

The morphological reconstruction process in the context of plant disease detection can be represented by the equation:

$$MF(f) = \Phi_{\text{rec}}(\delta(f), \varphi_k(f)) \quad \text{for } 0 \le k \le n$$

In this equation:

- MF(f) represents the morphologically reconstructed image, which highlights the areas of interest, such as diseased portions of plant leaves or stems.
- Φrec is the morphological reconstruction operator, which combines dilation and erosion processes to refine the image.
- δ(f) denotes the dilated image of the original image f, which emphasizes the features to be enhanced or examined.
- φk(f) is the reference image obtained after applying the closing operation k times to the pre-processed image. Closing is a morphological operation that helps in smoothening the contours of the objects, filling small holes, and connecting disjoint objects in the image.
- k represents the number of times the closing operation is applied, which is a parameter that can be adjusted based on the desired level of smoothening and connectivity in the image.
- n is the upper limit for k, dictating the maximum number of closing operations that can be applied, and is chosen based on the specific requirements of the image analysis task.

1.3. Multi-scale Gradient Image Generation technique

Applying the Multi-scale Gradient Image Generation technique for plant disease detection involves enhancing the edges of diseased areas on plant leaves or stems, which is crucial for accurate segmentation and subsequent identification of diseases. For plant disease detection, the multi-scale gradient MG(f) is computed by averaging the morphological gradients obtained using structuring elements Bi of different sizes. This approach captures the edges of diseased spots across multiple scales, ensuring that both small and large diseased areas are emphasized. The equation for the multi-scale gradient is:

$$MG(f) = \frac{1}{n}\sum_{i=1}^{n}[\partial(MF(f), B_i) - \varepsilon(MF(f), B_i)]$$

where n is the number of scales, and Bi is a structuring element of size (2i+1)×(2i+1). This process enhances the visibility of diseased spots by considering variations in their size and shape, which is particularly useful for detecting a wide range of plant diseases that manifest as spots or lesions of varying sizes (Rao et al., (2021), Vijaya Kumar et al., (2010)).

2. Marker Extraction in Plant Disease Detection

The primary goal of of Marker Extraction is to detect homogeneous regions within the plant image that correspond to diseased or healthy areas. The direct application of the Watershed algorithm to the gradient image of plant leaves can lead to over-segmentation due to noise and the complex textures of plant surfaces. Over-segmentation results in the division of the image into too many small segments, many of which may not correspond to actual diseased areas (Vijaya Kumar et al., (2010)).

2.1. Internal Markers for Plant Disease Detection

Identification of Internal Markers: Internal markers are identified within the objects of interest, which, in this case, are the diseased areas on the plant. These markers are obtained by eliminating extraneous minima that represent irrelevant details. This can be achieved through morphological operations that simplify the image, focusing on significant minima that correspond to actual diseased spots.

Role of Internal Markers: Internal markers specifically mark the diseased areas within the plant image. They help in accurately defining the boundaries of these areas for subsequent segmentation.

2.2. External Markers for Plant Disease Detection

Calculation of External Markers: External markers are calculated to represent the background or healthy tissue of the plant. These markers can be found by identifying pixels that are midway between the internal markers, effectively outlining the non-diseased areas.

Function of External Markers: External markers assist in differentiating between diseased and healthy areas, providing clear boundaries for the Watershed algorithm to segment the image effectively.

3.Watershed Segmentation Algorithm

The watershed segmentation algorithm is a powerful tool for image segmentation that treats

an image as a topographic surface, where the intensity of each pixel represents its height. The algorithm aims to find the "watershed lines" that separate distinct regions or "catchment basins" in the image (Lu et al., 2019). The main steps of the watershed segmentation algorithm are as follows:

1. Gradient Computation: Calculate the gradient magnitude of the input image using techniques such as the Sobel operator or morphological gradients. The gradient magnitude highlights the edges and boundaries in the image. The gradient magnitude G(x,y) at pixel (x,y) can be computed using the Sobel operator as:

$$G(x,y) = \sqrt{G_x^2(x,y) + G_y^2(x,y)}$$

where G_x and G_y are the horizontal and vertical gradient components, respectively

2. Marker Extraction: Identify markers or seed points within the image that represent the objects or regions of interest. These markers can be obtained through various techniques such as thresholding, morphological operations, or user-defined markers (Orbe-Trujillo et al., 2022). Let M(x,y) represent the marker image, where M(x,y)=1 for marker pixels and M(x,y)=0 for non-marker pixels.

3. Marker-Controlled Watershed: Perform the watershed transformation on the gradient image, using the extracted markers as the starting points. The markers act as the initial catchment basins, and the algorithm progressively grows these basins by following the gradient information until the entire image is segmented (Orbe-Trujillo et al., 2022). The marker-controlled watershed can be expressed as:

$$W(x,y) = \min_{(s,t)\in N(x,y)} \{W(s,t) + G(x,y)\}$$

where W(x,y) represents the watershed transform at pixel (x,y), N(x,y) denotes the neighborhood of pixel (x,y), and G(x,y) is the gradient magnitude at (x,y).

4. Region Merging: Optionally, merge adjacent regions based on similarity criteria to reduce over-segmentation. This step involves analyzing the properties of the segmented regions and merging them if they satisfy certain conditions, such as similar intensity, texture, or shape (Roshni & Raju (2011))

The watershed segmentation algorithm relies on mathematical concepts from morphological image processing, such as erosion, dilation, and reconstruction. These morphological operations are defined using set theory and mathematical expressions. For example, the erosion of an image f by a structuring element B is given by:

$$(f \ominus B)(x,y) = \min_{(s,t)\in B} \{f(x+s, y+t)\}$$

Similarly, the dilation of an image f by a structuring element B is defined as:

$$(f \oplus B)(x,y) = \max_{(s,t)\in B} \{f(x-s, y-t)\}$$

4.Hu Moments Features

Hu Moments are mathematical descriptors used in image processing to capture the shape of objects within an image, invariant to transformations such as translation, scale, and rotation. These moments are derived from the theory of moments in image analysis, providing a robust feature set for pattern recognition and shape analysis. The Hu Moments are calculated through a series of mathematical steps, starting from raw image moments, moving to central moments, normalized central moments, and finally resulting in the seven invariant Hu Moments. Here's how they are mathematically derived:

Raw Moments: The foundation of Hu Moments is the raw image moments, which are the sum of pixel intensities raised to the power of their coordinates:

$$M_{pq} = \sum_x \sum_y x^p y^q f(x,y)$$

where f(x,y) represents the pixel intensity at coordinates (x,y) and p and q are the orders of the moment.

Central Moments: Central moments, which are translation invariant, are computed as:

$$\mu_{pq} = \sum_x \sum_y (x-\bar{x})^p (y-\bar{y})^q f(x,y)$$

where $\bar{x} = \frac{M_{10}}{M_{00}}$ and $\bar{y} = \frac{M_{01}}{M_{00}}$ are the coordinates of the centroid of the image.

Normalized Central Moments: To achieve scale invariance, these central moments are normalized:

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^{\gamma}}$$

where $\gamma = (p+q)/2 + 1$, ensuring that the moments are independent of the object's size.

Hu Moments

Hu introduced seven moment invariants, which are combinations of normalized central moments:

- $\mathrm{Hu}_1 = \eta_{20} + \eta_{02}$
- $\mathrm{Hu}_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2$
- $\mathrm{Hu}_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2$
- $\mathrm{Hu}_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2$
- $\mathrm{Hu}_5 = (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]$
- $\mathrm{Hu}_6 = (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03})$
- $\mathrm{Hu}_7 = (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] - (\eta_{30} - 3\eta_{12})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]$

These seven Hu Moments are used as features in image processing and computer vision for tasks such as object recognition, shape analysis, and pattern matching. The invariance of Hu Moments to image transformations makes them extremely valuable in scenarios where the object's orientation, scale, or position varies (Xie et al., (2020)).

- Pattern Recognition: Hu Moments help in identifying objects in images regardless of their orientation or size, which is crucial for automated image classification systems.

- Shape Analysis: They provide a compact but comprehensive representation of an object's shape, facilitating detailed shape analysis and comparison in tasks like object detection and tracking.

Hu Moments stand out in image analysis for their robustness to changes in scale, position, and rotation, making them a cornerstone feature for shape-based image recognition and classification.

5. Gray Level Co-occurrence Matrix (GLCM)

The Gray Level Co-occurrence Matrix (GLCM) is a statistical method used to examine the texture of an image by considering the spatial relationship of pixels. It is particularly useful for classifying types of diseases in tomato plants by analyzing the texture on the leaf surface (Sutarno & Putri Fauliah. (2019); Sinaga et al., (2021); Khan et al., (2021); Abolghasemi et al., (2008)). The algorithm steps for GLCM and the features derived from it are as follows:

Algorithm Steps:

1. Slide a window across the image or consider the whole image to calculate the frequency of pixel intensity co-occurrences at specific distances and angles.

2. Construct the GLCM for each position of the window.

3. Extract statistical measures from the GLCM, such as contrast, correlation, energy, and homogeneity, which serve as texture features.

GLCM Features:

1. Contrast: Measures the intensity contrast between a pixel and its neighbor over the whole image.

$$\text{Contrast} = \sum_{i=0}^{levels-1} \sum_{j=0}^{levels-1} (i-j)^2 P(i,j)$$

2. Homogeneity: Measures the closeness of the distribution of elements in the GLCM to the GLCM diagonal.

$$\text{Homogeneity} = \sum_{i=0}^{levels-1} \sum_{j=0}^{levels-1} \frac{P(i,j)}{1+(i-j)^2}$$

3. Correlation: Measures how correlated a pixel is to its neighbor over the whole image.

$$\text{Correlation} = \sum_{i=0}^{levels-1} \sum_{j=0}^{levels-1} \frac{(i-\mu_i)(j-\mu_j)P(i,j)}{\sigma_i \sigma_j}$$

where μ_i, μ_j are the means and σ_i, σ_j are the standard deviations of the row and column sums of the GLCM, respectively.

4. Energy (or Angular Second Moment): Provides the sum of squared elements in the GLCM.

$$\text{Energy} = \sum_{i=0}^{levels-1} \sum_{j=0}^{levels-1} P(i,j)^2$$

6. Gabor Features

Gabor features are utilized to analyze the texture and edge information of images, providing valuable insights for identifying disease patterns in tomato leaves. The study employs a Gabor filter defined by specific parameters to capture these characteristics effectively. The Gabor filter's parameters are crucial as they determine the nature of the texture and edge information that will be extracted. The Kernel Size (ksize) is set to (10, 10), which defines the dimensions of the filter. This size is selected to balance between capturing sufficient local detail and maintaining computational efficiency. A larger kernel would capture broader features but might miss finer details, whereas a smaller kernel would focus on finer details but might miss the broader context. The Sigma (sigma) is set to 5, controlling the width of the Gaussian envelope of the Gabor function. This parameter influences the scale of the features that the filter responds to. A larger sigma would capture more global features, while a smaller sigma would focus on more localized details. In this study, a sigma of 5 is chosen to capture a balanced range of textures. The Theta (theta) is set to 1 radian, determining the orientation of the normal to the parallel stripes of the Gabor function. This orientation sensitivity allows the filter to detect edges and textures at a specific angle. Setting theta to 1 radian (approximately 57.3 degrees) means the filter is particularly tuned to capture features aligned along this orientation, which can be critical for identifying certain disease patterns that manifest in specific directions. The Lambda (lambda) is set to 0.6, representing the wavelength of the sinusoidal factor. This parameter defines the frequency of the sinusoidal wave within the Gabor function. A lower lambda focuses on higher frequency components, capturing finer textures, while a higher lambda captures coarser textures. The chosen value of 0.6 strikes a balance to capture a range of texture details pertinent to leaf disease detection. The Gamma (gamma) is set to 1, indicating the spatial aspect ratio. This parameter controls the ellipticity of the Gabor filter. A gamma of 1 means the filter is circular, which is appropriate for capturing isotropic textures where features are equally likely in all directions. This helps in detecting uniform textures that are common in diseased leaf areas. The Psi (psi) is set to 0, which is the phase offset of the sinusoidal function. This parameter shifts the sinusoidal wave within the

Gabor function. A phase offset of 0 means the wave starts at zero, making the filter symmetric. This symmetry is useful for capturing balanced texture patterns without introducing directional bias. The Gabor filter is a powerful tool for texture analysis in image processing, particularly useful for feature extraction in disease classification. It captures texture and edge information that can be indicative of plant diseases (Vatamanu & Ionescu, 2012; Boughida et al., 2021; Hajraoui & Sabri, 2014; Chuan et al., 2013). The algorithm steps for using Gabor filters and the equation for Gabor feature extraction are as follows: First, design a set of Gabor filters with various frequencies and orientations. Next, convolve the image with each filter to obtain a response that represents local texture information. Finally, analyze the filter responses to extract features that are useful for the classification of diseases in tomato plants. Gabor filters are used to extract texture features at different frequencies and orientations. A Gabor filter is a linear filter used for texture analysis, which means it analyzes whether there are any specific frequency content in the image in specific directions in a localized region around the point or region of analysis. The Gabor feature extraction can be represented by the following equation:

$$G(x, y; \lambda, \theta, \psi, \sigma, \gamma) = \exp\left(-\frac{x'^2+\gamma^2 y'^2}{2\sigma^2}\right) \cos\left(2\pi\frac{x'}{\lambda} + \psi\right)$$

where

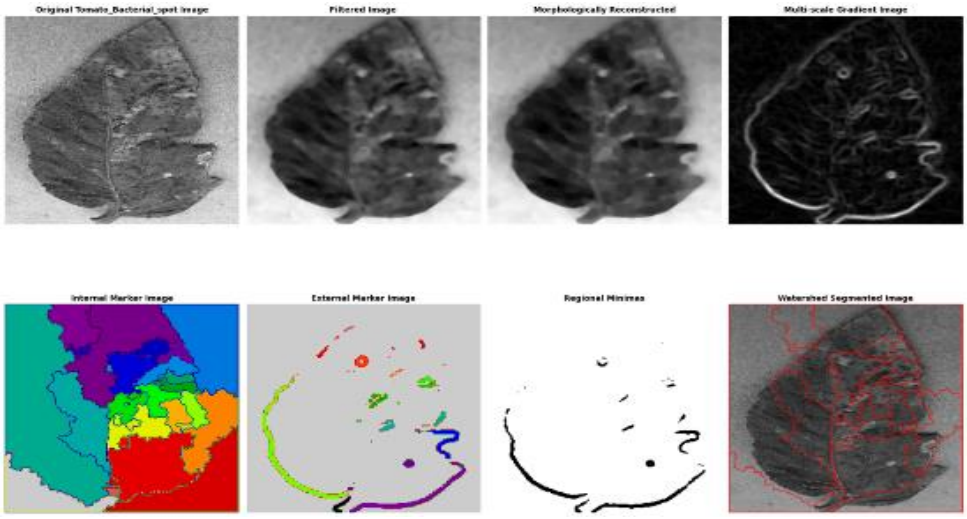
- $x' = x\cos\theta + y\sin\theta$,
- $y' = -x\sin\theta + y\cos\theta$,
- λ is the wavelength of the sinusoidal factor,
- θ is the orientation of the normal to the parallel stripes of a Gabor function,
- ψ is the phase offset,
- σ is the sigma/standard deviation of the Gaussian envelope, and
- γ is the spatial aspect ratio.

These equations for GLCM and Gabor features form the mathematical basis for extracting the textural and pattern information necessary for classifying the leaf images into healthy or diseased categories. By applying these techniques, the study aims to improve the accuracy of detecting and categorizing diseases in tomato plants, leveraging the power of image processing and machine learning algorithms.

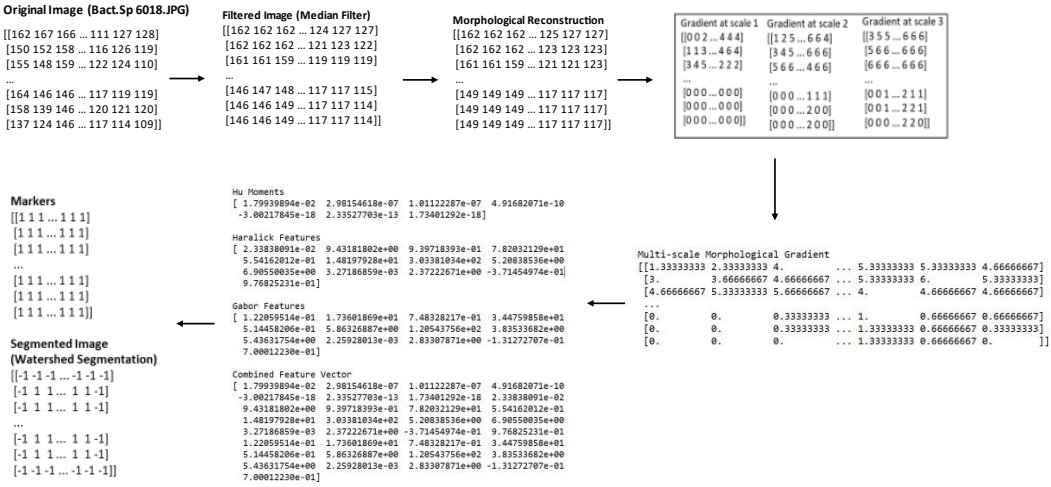
2. Discussion

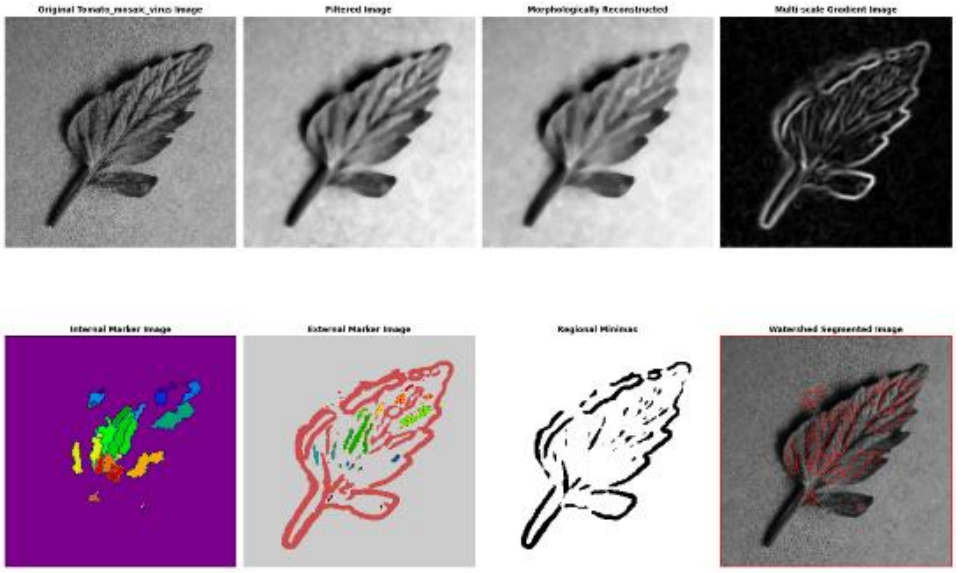
The study aimed to enhance the accuracy of tomato plant disease classification by employing advanced image processing techniques and machine learning algorithms. Upon comparing the existing and proposed methods, significant improvements were observed in the proposed approach (Fig.1). The existing method utilized median filtering for noise reduction and morphological reconstruction to enhance features, followed by multi-scale gradient image generation. Feature extraction included hue moments and Haralick features. Classification was performed using various machine learning algorithms, including Support Vector Machines (SVM), Random Forest Classifier (RFC), Logistic Regression, and XGBoost. The SVM

classifier achieved an accuracy of 91%, with other classifiers showing similar performance levels around 90-91%. In contrast, the proposed method introduced additional preprocessing steps, such as the application of internal and external markers for precise segmentation using the watershed algorithm, which significantly resolved the over-segmentation issue prevalent in earlier methods. Moreover, the inclusion of Gabor features for texture analysis, alongside hue moments and Haralick features, resulted in a more robust feature extraction process. This combination of textural and shape information enriched the dataset, allowing for a more nuanced differentiation between disease presentations on tomato leaves. The proposed method demonstrated notable improvements in classification performance. The SVM classifier's accuracy increased to 93%, Random Forest accuracy increased from 90% to 91%, Logistic Regression accuracy increased from 90% to 92%, and XGBoost accuracy increased from 91% to 92%. These improvements underscore the proposed method's refined ability to discriminate between healthy and diseased tomato plants, crucial for preventing crop losses (Fig.2; Table.1; Table.3; Graph.1). Upon comparing the classification reports, the proposed method showed enhancements across several key metrics: Precision for identifying 'Tomato Bacterial Spot' increased from 0.87 to 0.90, indicating fewer false positives. Recall for 'Tomato Yellow Leaf Curl Virus' rose from 0.95 to 0.96, reflecting better identification of relevant cases. F1-score for 'Tomato Mosaic Virus' increased from 0.64 to 0.71, suggesting more reliable performance despite a small support size. Overall, the proposed method achieved a higher average F1-score of 0.88 compared to 0.85 in the existing method. The overall accuracy of the system increased from 0.91 to 0.93, with similar improvements observed in macro and weighted averages of precision, recall, and F1-score. These metrics indicate a generally more balanced and effective classification system (Table.2; Graph.2). The advancements in the proposed method have significant implications for agricultural disease management. Enhanced detection of diseases such as Bacterial Spot, Tomato Mosaic Virus, and Tomato Yellow Leaf Curl Virus enables more reliable and timely implementation of treatment strategies. The proposed method may also offer a generalized framework adaptable to other forms of agricultural disease detection, paving the way for broad-spectrum applications.

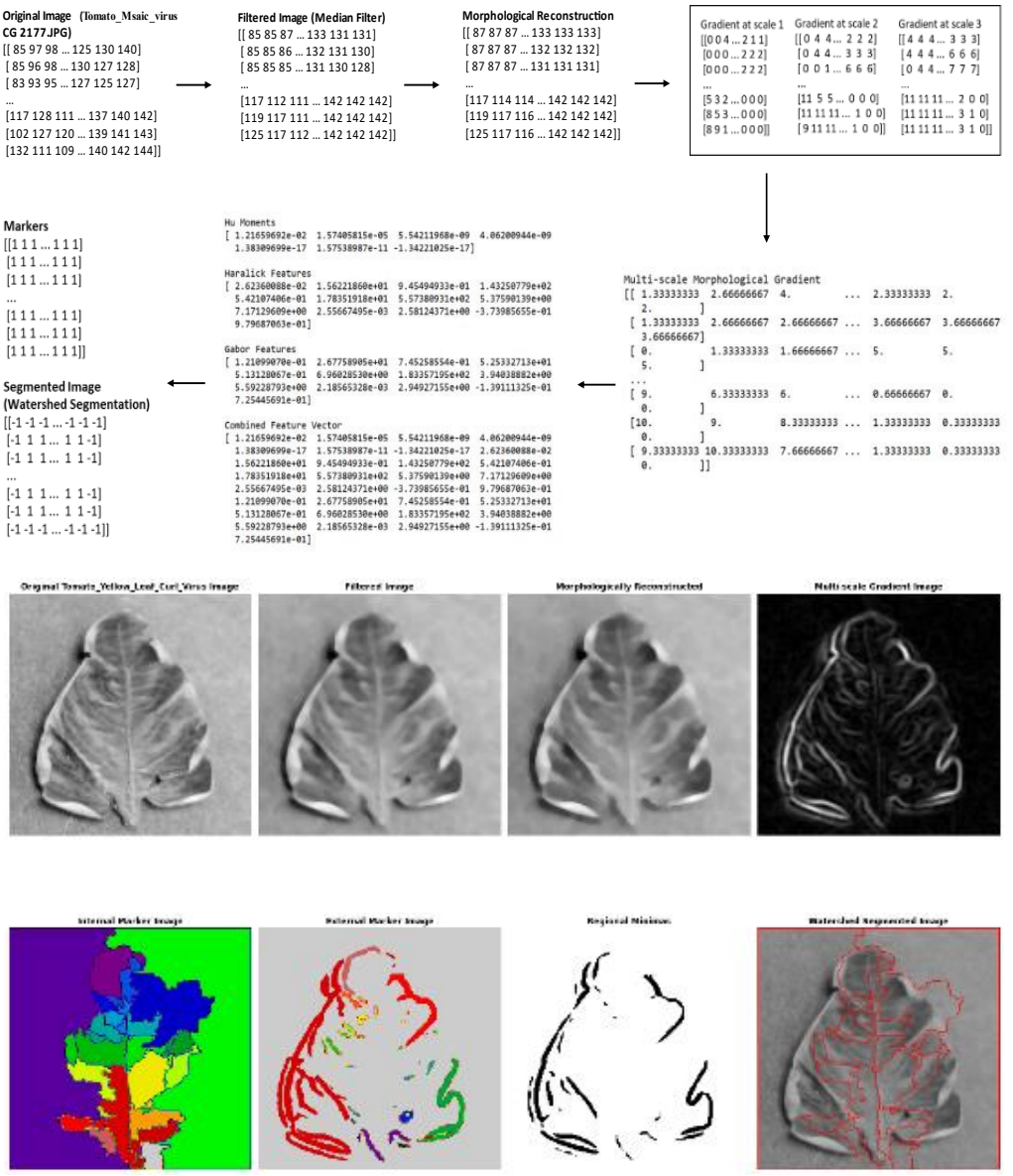


(a) Tamoto Bacterial_Spot





(b) Tomato_Msaic_virus



(c) Tomato_Yearly_leaf_curl_virus

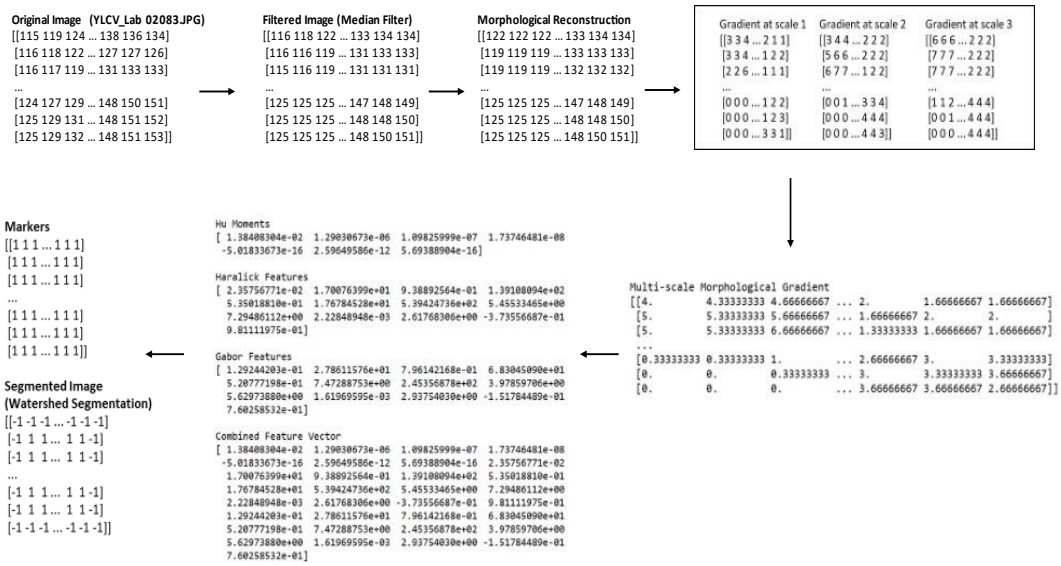


Fig.2. (a) Tamoto Bacterial_Spot , (b) Tomato_Msaic_virus, (c) Tomato_Yearly_leaf_curl_virus

Table.1. Hue,Haralic and Gabor features of Tomato Diseases (Tomato Bcterial spot, Tomato Mosaic Virus, and Tomato Yellow leaf curl virus)

Moment_1	Moment_2	Moment_3	Moment_4	Moment_5	Moment_6	Moment_7	Haralick_1	Haralick_2	Haralick_3	Haralick_4	Haralick_5	Haralick_6	Haralick_7	Haralick_8	Haralick_9	Haralick_10
0.020808727	1.59E-05	6.04E-07	8.83E-10	-4.73E-19	-9.83E-13	2.04E-17	0.035014669	13.20418802	0.937881403	106.2355668	0.612080942	13.42604933	411.7380792	4.995921485	6.447660874	0.002981973
0.020253395	9.79E-06	5.89E-08	2.99E-08	8.72E-16	4.78E-12	-9.04E-16	0.024914364	8.597743101	0.94221438	74.36273583	0.584477441	14.23426875	288.8532002	5.124628595	6.689294569	0.003631397
0.014798793	3.89E-06	5.80E-08	1.84E-08	5.88E-16	-7.93E-12	1.15E-16	0.031335955	10.6207922	0.945132609	96.74951508	0.578317895	15.40600158	376.3772681	5.228725636	6.849488279	0.002712951
0.015295848	1.36E-06	6.40E-08	7.05E-09	-7.90E-17	-6.13E-12	1.27E-16	0.017894055	11.56397928	0.939532134	95.5907244	0.528877978	17.62366393	370.7989183	5.510165319	7.382690965	0.002567783
0.018112416	9.62E-07	7.85E-08	3.20E-08	1.38E-15	-3.33E-12	-8.17E-16	0.023500872	19.49433487	0.937901762	156.8925222	0.560802463	16.40599832	608.0757539	5.256971099	6.944399255	0.00230442
0.019331331	4.73E-06	2.86E-07	1.57E-07	1.81E-14	3.31E-10	2.81E-14	0.027627709	19.47099915	0.930092209	139.1993863	0.580380023	14.81742736	537.3265461	5.102642943	6.685331623	0.002328024
0.026736544	8.78E-07	1.73E-08	5.12E-07	-1.65E-14	8.71E-11	4.53E-14	0.032542068	6.481041967	0.940963823	54.86831414	0.61101739	11.61677294	212.9922146	4.746818579	6.134808534	0.003869747
0.021335444	2.70E-06	1.62E-07	2.98E-09	-6.42E-19	-4.26E-12	6.53E-17	0.031838916	9.757898389	0.945531194	89.53532314	0.60381303	13.56016987	348.3833942	5.032995678	6.508387675	0.00314411
0.020297946	1.63E-06	2.80E-07	6.69E-08	8.78E-15	2.00E-11	2.59E-15	0.024364876	10.05371073	0.938666267	81.92895248	0.565480975	14.70987596	317.6620992	5.208162025	6.852741453	0.002991723

Haralick_11	Haralick_12	Haralick_13	Gabor_1	Gabor_2	Gabor_3	Gabor_4	Gabor_5	Gabor_6	Gabor_7	Gabor_8	Gabor_9	Gabor_10	Gabor_11	Gabor_12	Gabor_13	Label
2.224293605	-0.404174879	0.979254919	0.126742	22.42775	0.781261	51.244	0.531481	5.836329	182.5483	3.691118	5.189044	0.002477	2.75347	-0.14104	0.705887	Tomato___Bacterial_spot
2.219378391	-0.390522839	0.978874866	0.123882	16.1196	0.78826	38.04797	0.528892	5.5383	136.0723	3.730907	5.265064	0.002314	2.73277	-0.13844	0.705983	Tomato___Bacterial_spot
2.411388263	-0.395865839	0.981663971	0.123087	18.13729	0.747837	35.94851	0.520366	5.928157	125.6567	3.800982	5.378253	0.002673	2.817447	-0.13847	0.717089	Tomato___Bacterial_spot
2.545870249	-0.372691273	0.981466853	0.118327	21.46108	0.792682	51.7417	0.505231	6.934408	185.5057	3.99023	5.660369	0.002089	2.928093	-0.14572	0.741157	Tomato___Bacterial_spot
2.411245087	-0.379183396	0.978924513	0.122221	42.7362	0.788321	100.9197	0.518616	7.402044	360.9427	3.920828	5.52829	0.001461	2.911544	-0.14969	0.743812	Tomato___Bacterial_spot
2.333235408	-0.382261684	0.976865008	0.122099	41.60965	0.662862	61.72851	0.521369	6.332254	205.3044	3.809557	5.37695	0.001813	2.871415	-0.13277	0.69185	Tomato___Bacterial_spot
2.032456617	-0.379556883	0.969412473	0.120712	12.01863	0.686868	19.18575	0.529436	4.488536	64.72437	3.566379	5.048696	0.003482	2.638787	-0.11661	0.631958	Tomato___Bacterial_spot
2.194291826	-0.401816369	0.979467613	0.128622	13.67373	0.783999	31.6387	0.536427	5.224812	112.8811	3.659933	5.148636	0.003024	2.676678	-0.13879	0.7016	Tomato___Bacterial_spot
2.341346338	-0.384266407	0.978917675	0.121641	17.00533	0.675988	26.24613	0.521969	5.437831	87.97917	3.776586	5.351762	0.003192	2.778006	-0.1275	0.676304	Tomato___Bacterial_spot

Moment_1	Moment_2	Moment_3	Moment_4	Moment_5	Moment_6	Moment_7	Haralick_1	Haralick_2	Haralick_3	Haralick_4	Haralick_5	Haralick_6	Haralick_7	Haralick_8	Haralick_9	Haralick_10
0.012379432	2.88E-06	1.67E-08	7.01E-09	6.26E-17	1.19E-11	4.28E-17	0.028164414	13.64507605	0.94509022	124.1975119	0.561673769	16.30455576	483.1449717	5.245448737	6.936749562	0.002660769
0.014350008	4.44E-06	8.53E-09	7.11E-10	-5.61E-19	-1.49E-12	1.66E-18	0.020877775	11.53723382	0.941614091	98.76336204	0.535731529	16.34059472	383.5162143	5.357968984	7.159132482	0.00281995
0.009628363	2.36E-06	2.90E-08	4.72E-09	-5.18E-17	-5.26E-12	-1.93E-17	0.019853039	38.0706792	0.934049794	288.5044383	0.504546317	22.33671498	1115.947074	5.654548196	7.707900776	0.001778074
0.011117641	4.74E-08	7.39E-09	2.26E-09	4.20E-18	-1.13E-13	8.25E-18	0.013145701	25.96046165	0.935425124	200.9176615	0.468295418	20.60416993	777.7101844	5.583589237	7.705149232	0.001692828
0.013115449	3.54E-06	3.48E-09	1.07E-08	-3.66E-17	-6.17E-12	5.45E-17	0.022807487	23.70291231	0.938370905	192.2086814	0.542533127	17.15164852	745.1318133	5.222027026	7.008796369	0.002393684
0.011720984	2.40E-06	1.15E-08	1.53E-08	4.55E-18	-1.90E-11	2.02E-16	0.020278927	25.08436624	0.937495852	200.5695316	0.519012971	19.19287157	777.1937601	5.454612844	7.381718518	0.002203042
0.01204448	3.72E-06	4.66E-09	5.59E-09	2.12E-17	-1.49E-12	1.91E-17	0.017661138	18.75220367	0.940083197	156.4174381	0.505665208	18.37994478	606.9175486	5.430353122	7.367997143	0.002009682
0.009779203	1.54E-06	4.18E-09	1.69E-08	9.42E-17	-2.09E-11	-1.06E-16	0.017803385	38.22195554	0.939049554	313.4503458	0.497541004	22.75180134	1215.579428	5.688897837	7.751976697	0.001620389
0.012297888	6.20E-06	2.00E-08	2.66E-09	4.54E-19	6.59E-12	1.94E-17	0.018405427	25.1290661	0.938218529	203.2759521	0.504253896	19.33751147	787.9747422	5.447029787	7.433316953	0.001991188

Haralick_11	Haralick_12	Haralick_13	Gabor_1	Gabor_2	Gabor_3	Gabor_4	Gabor_5	Gabor_6	Gabor_7	Gabor_8	Gabor_9	Gabor_10	Gabor_11	Gabor_12	Gabor_13	Label
2.461469064	-0.379623368	0.978957288	0.118907	23.15244	0.785521	53.95	0.514048	6.418729	192.6475	3.827004	5.417013	0.002173	2.877586	-0.13675	0.711995	Tomato___Tomato_mosaic_virus
2.481428515	-0.368435616	0.97859048	0.116832	22.40821	0.770692	48.84054	0.506505	6.447054	172.954	3.913145	5.564017	0.001892	2.918815	-0.13151	0.708127	Tomato___Tomato_mosaic_virus
2.886112797	-0.351839317	0.979730829	0.124029	59.60195	0.801043	149.7197	0.502227	10.22302	539.2767	4.179978	5.924588	0.001446	3.188666	-0.15452	0.772494	Tomato___Tomato_mosaic_virus
2.770535396	-0.328731071	0.973663787	0.106627	40.47274	0.754782	82.48935	0.477577	8.142249	289.4846	4.158935	5.962846	0.001203	3.147471	-0.12852	0.71675	Tomato___Tomato_mosaic_virus
2.531618536	-0.351563003	0.972713307	0.123499	37.79597	0.798551	93.76805	0.51593	7.794621	337.2762	3.952669	5.594227	0.001411	2.967275	-0.14566	0.740375	Tomato___Tomato_mosaic_virus
2.68775486	-0.35312396	0.977128722	0.125302	44.08316	0.814154	118.5471	0.504036	8.960004	430.1053	4.087664	5.788835	0.001446	3.097886	-0.15176	0.76172	Tomato___Tomato_mosaic_virus
2.621693073	-0.347484235	0.975371872	0.115906	37.75174	0.806566	97.53755	0.498405	7.809201	352.3985	4.013793	5.701946	0.001274	3.039546	-0.13938	0.731923	Tomato___Tomato_mosaic_virus
2.89217749	-0.355179799	0.980951028	0.128984	61.45911	0.779143	139.1437	0.505644	10.11736	495.1155	4.196347	5.959647	0.001439	3.196294	-0.1519	0.770367	Tomato___Tomato_mosaic_virus
2.716984378	-0.338997181	0.973719713	0.123305	41.39432	0.778043	99.20801	0.502998	8.474152	331.4377	4.091984	5.820845	0.001619	3.087193	-0.14379	0.745339	Tomato___Tomato_mosaic_virus

Moment_1	Moment_2	Moment_3	Moment_4	Moment_5	Moment_6	Moment_7	Haralick_1	Haralick_2	Haralick_3	Haralick_4	Haralick_5	Haralick_6	Haralick_7	Haralick_8	Haralick_9	Haralick_10
0.015881347	2.11E-06	5.61E-09	6.97E-09	4.11E-17	-8.54E-12	-1.47E-17	0.022533001	7.851288537	0.949048867	77.01747116	0.568408106	14.87066972	300.2185961	5.191014829	6.812996578	0.003530923
0.013416039	9.87E-08	1.27E-08	4.34E-08	7.39E-16	-1.36E-11	-7.05E-16	0.019732831	9.536539657	0.941891482	82.03000841	0.53271148	16.46697615	318.583494	5.37293293	7.181165663	0.002953957
0.009040152	1.05E-06	1.21E-08	1.05E-09	-3.18E-18	3.38E-13	1.97E-18	0.009541814	31.27351394	0.933027529	233.4451429	0.417989103	27.60778697	902.5070576	6.126848091	8.616299812	0.00157642
0.013606679	5.98E-07	1.04E-07	9.72E-09	1.96E-16	6.44E-12	2.39E-16	0.012383156	8.669436905	0.930103655	62.00756937	0.483225886	18.49328314	239.3608406	5.577818942	7.619991968	0.003433206
0.011142738	4.68E-07	8.42E-08	6.41E-10	-4.67E-18	-4.38E-13	-5.53E-19	0.022928885	18.89080307	0.940858714	159.6510905	0.531033655	19.62175973	619.7135589	5.594767832	7.505303974	0.002006911
0.007767272	1.13E-07	1.21E-08	5.41E-09	3.08E-17	-1.82E-12	-3.11E-17	0.016522977	26.43871185	0.953320346	283.0993013	0.468875908	27.81430147	1105.958493	6.080324052	8.308135716	0.001310105
0.009813863	1.36E-06	1.18E-08	7.92E-09	-7.55E-17	-2.60E-12	-1.28E-17	0.012168848	28.15250281	0.939183608	231.3931544	0.450721789	25.41758943	897.4201149	5.953386718	8.28082992	0.001910205
0.011275121	2.69E-07	1.33E-09	1.39E-08	-4.40E-17	-3.78E-12	4.01E-17	0.01051404	26.184319	0.941283603	222.8748649	0.442467128	23.68823351	865.3151404	5.792802031	8.087777165	0.00167102
0.007730572	3.50E-07	1.65E-09	1.10E-09	1.25E-18	-3.54E-13	7.87E-19	0.011124406	35.32465437	0.940367793	296.0770713	0.432825372	28.52239016	1148.983631	6.093855059	8.52407413	0.001474851

Haralick_11	Haralick_12	Haralick_13	Gabor_1	Gabor_2	Gabor_3	Gabor_4	Gabor_5	Gabor_6	Gabor_7	Gabor_8	Gabor_9	Gabor_10	Gabor_11	Gabor_12	Gabor_13	Label
2.239492692	-0.38647204	0.979085538	0.120724	15.11858	0.776716	33.84154	0.524309	5.659663	120.2476	3.781386	5.356219	0.002394	2.744255	-0.13769	0.712381	Tomato___Tomato_Yellow_Leaf_Curl_Virus
2.460083224	-0.369015168	0.978922771	0.110578	17.47656	0.735275	32.99581	0.502214	6.009243	114.5067	3.901751	5.565517	0.002019	2.877779	-0.12627	0.693573	Tomato___Tomato_Yellow_Leaf_Curl_Virus
3.162780889	-0.325485402	0.980788959	0.128763	53.483	0.800981	134.3609	0.487239	12.01713	483.9604	4.465114	6.406419	0.001481	3.322754	-0.16216	0.809589	Tomato___Tomato_Yellow_Leaf_Curl_Virus
2.594620947	-0.344772571	0.976402589	0.096283	17.07793	0.590148	20.82793	0.471922	6.017221	66.2338	4.004025	5.768655	0.003438	2.967716	-0.10382	0.620404	Tomato___Tomato_Yellow_Leaf_Curl_Virus
2.700768674	-0.37724278	0.983289249	0.118139	23.23571	0.763233	49.04808	0.509342	7.184142	172.9566	4.023493	5.717404	0.002604	2.950044	-0.14526	0.739499	Tomato___Tomato_Yellow_Leaf_Curl_Virus
3.039455142	-0.37296248	0.987828143	0.115045	36.46	0.817036	99.5939	0.489106	9.742723	361.9156	4.306924	6.137096	0.001247	3.161315	-0.16292	0.794988	Tomato___Tomato_Yellow_Leaf_Curl_Virus
3.040969486	-0.335406826	0.980571276	0.124922	47.66681	0.794815	116.1141	0.486653	10.68679	416.7894	4.363021	6.241772	0.001422	3.280592	-0.15268	0.783759	Tomato___Tomato_Yellow_Leaf_Curl_Virus
2.911669053	-0.321094767	0.975060375	0.116223	38.72961	0.775405	86.18469	0.47797	9.203353	306.0092	4.290268	6.159112	0.001252	3.216214	-0.13912	0.749506	Tomato___Tomato_Yellow_Leaf_Curl_Virus
3.166898563	-0.333855969	0.982052819	0.122569	57.04515	0.790452	136.0571	0.480606	11.65731	487.1834	4.448565	6.380724	0.001347	3.356725	-0.15316	0.792599	Tomato___Tomato_Yellow_Leaf_Curl_Virus

Table.2. (a) & (b) Classification Report for Existing and Proposed methods

(a) Classification Report for SVM: Existing Method

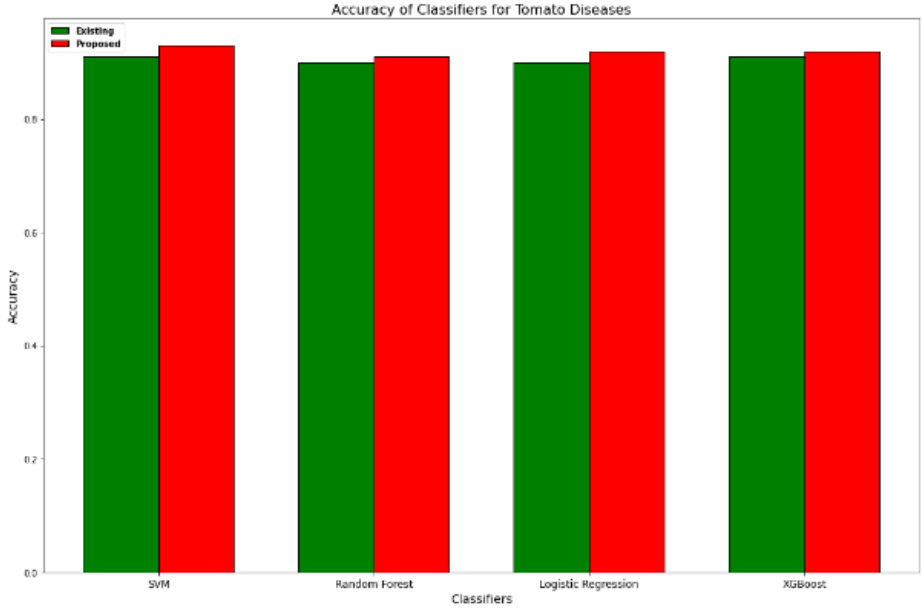
	Precision	Recall	F1-score	Support
Tomato___Bacterial_spot	0.87	0.82	0.85	425
Tomato___Tomato_Yellow_Leaf_Curl_Virus	0.90	0.95	0.92	1069
Tomato___Tomato_mosaic_virus	0.88	0.51	0.64	69
Tomato___healthy	0.98	0.97	0.97	327
Accuracy			0.91	1890
macro avg	0.91	0.81	0.85	1890
weighted avg	0.91	0.91	0.90	1890

(b) Classification Report for SVM: Proposed Method

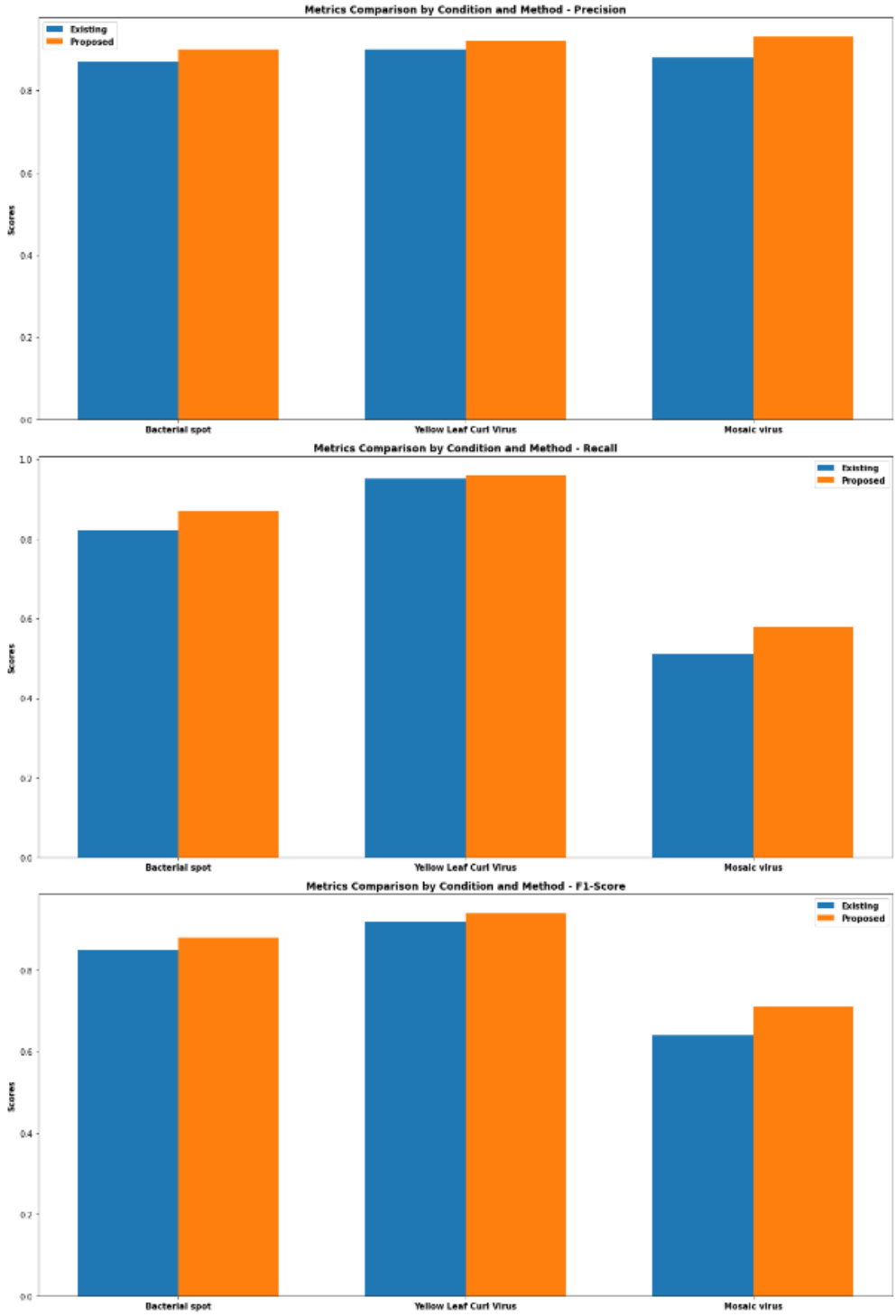
	Precision	Recall	F1-score	Support
Tomato___Bacterial_spot	0.90	0.87	0.88	425
Tomato___Tomato_Yellow_Leaf_Curl_Virus	0.92	0.96	0.94	1069
Tomato___Tomato_mosaic_virus	0.93	0.58	0.71	69
Tomato___healthy	0.99	0.97	0.98	327
Accuracy			0.93	1890
macro avg	0.94	0.85	0.88	1890
weighted avg	0.93	0.93	0.93	1890

Table.3. Accuracies of the Classifiers for Existing and Proposed method

Classifier	Existing Accuracy	Proposed Accuracy
SVM	91%	93%
Random Forest	90%	91%
Logistic Regression	90%	92%
XGBoost	91%	92%



Graph.1. Accuracy of classifiers for Tomato Diseases



Graph.2. Comparison metrics of the Existing and Proposed methods

3. Conclusion

This study significantly enhances the accuracy of tomato plant disease classification by employing advanced image processing techniques and machine learning algorithms. The proposed method introduced precise segmentation using the watershed algorithm and enriched feature extraction with Gabor features, hue moments, and Haralick features. These improvements led to more accurate and nuanced disease classification. Key results include an increase in classification accuracy across multiple classifiers: SVM accuracy rose from 91% to 93%, Random Forest from 90% to 91%, Logistic Regression from 90% to 92%, and XGBoost from 91% to 92%. The proposed method also showed better precision, recall, and F1-scores for various diseases, leading to an overall system accuracy improvement from 91% to 93%. In conclusion, the proposed method significantly improves upon the existing method, demonstrating the benefits of integrating advanced image processing techniques and enriched feature extraction methods in machine learning models.

References

1. Abrahamian, P., Klein-Gordon, J. M., Jones, J. B., & Vallad, G. E. (2021). Epidemiology, diversity, and management of bacterial spot of tomato caused by Xanthomonas perforans. Applied microbiology and biotechnology, 105(16-17), 6143–6158. https://doi.org/10.1007/s00253-021-11459-9.
2. Potnis, N., Timilsina, S., Strayer, A., Shantharaj, D., Barak, J. D., Paret, M. L., Vallad, G. E., & Jones, J. B. (2015). Bacterial spot of tomato and pepper: diverse Xanthomonas species with a wide variety of virulence factors posing a worldwide challenge. Molecular plant pathology, 16(9), 907–920. https://doi.org/10.1111/mpp.12244
3. Shen, X., Yan, Z., Wang, X., Wang, Y., Arens, M., Du, Y., Visser, R. G. F., Kormelink, R., Bai, Y., & Wolters, A. A. (2020). The NLR Protein Encoded by the Resistance Gene Ty-2 Is Triggered by the Replication-Associated Protein Rep/C1 of Tomato Yellow Leaf Curl Virus. Frontiers in plant science, 11, 545306. https://doi.org/10.3389/fpls.2020.545306
4. Xian, T. and Ngadiran, R. (2021). plant diseases classification using machine learning. Journal of Physics Conference Series, 1962(1), 012024. https://doi.org/10.1088/1742-6596/1962/1/012024
5. Jos, J. and Venkatesh, K. (2020). disease detection in plants using a pseudo color co occurrence matrix. International Journal of Engineering and Advanced Technology, 9(4), 1485-1490. https://doi.org/10.35940/ijeat.d7488.049420
6. Geetha, G., Samundeswari, S., Saranya, G., Meenakshi, K., & Nithya, M. (2020). plant leaf disease classification and detection system using machine learning. Journal of Physics Conference Series, 1712(1), 012012. https://doi.org/10.1088/1742-6596/1712/1/012012
7. Nasir, M. and Fajri, M. (2019). Identification of diseases in rice plants using the gray level co-occurrence matrix method. Iop Conference Series Materials Science and Engineering, 536(1), 012146. https://doi.org/10.1088/1757-899x/536/1/012146
8. Arjunagi, S. and Patil, N. (2019). Texture based leaf disease classification using machine learning techniques. International Journal of Engineering and Advanced Technology, 9(1), 956-961. https://doi.org/10.35940/ijeat.a9446.109119
9. Samrin, R. (2021). Weed classification on images using machine learning. International Journal for Research in Applied Science and Engineering Technology, 9(VI), 1221-1227. https://doi.org/10.22214/ijraset.2021.35173
10. Jeyalakshmi, S. and Radha, R. (2020). An effective approach to feature extraction for

classification of plant diseases using machine learning. Indian Journal of Science and Technology, 13(32), 3295-3314. https://doi.org/10.17485/ijst/v13i32.827

11. Manik, F., Saputra, S., & Ginting, D. (2020). plant classification based on extraction feature gray level co-occurrence matrix using k-nearest neighbour. Journal of Physics Conference Series, 1566(1), 012107. https://doi.org/10.1088/1742-6596/1566/1/012107
12. Sutarno, & Putri Fauliah, S. (2019). Implementation of Learning Vector Quantization (LVQ) Algorithm for Durian Fruit Classification Using Gray Level Co-occurrence Matrix (GLCM) Parameters. Journal of Physics: Conference Series, 1196.
13. Sinaga, D., Agustina, F., Setiyanto, N.A., Suprayogi, S., & Jatmoko, C. (2021). Classification of Bird Based on Face Types Using Gray Level Co-Occurrence Matrix (GLCM) Feature Extraction Based on the k-Nearest Neighbor (K-NN) Algorithm. Journal of Applied Intelligent System.
14. Khan, S.U., Islam, N., Jan, Z., Haseeb, K., Shah, S.I., & Hanif, M. (2021). A machine learning-based approach for the segmentation and classification of malignant cells in breast cytology images using gray level co-occurrence matrix (GLCM) and support vector machine (SVM). Neural Computing and Applications, 34, 8365 - 8372.
15. Abolghasemi, M., Aghainia, H., Faez, K., & Mehrabi, M.A. (2008). LSB data hiding detection based on gray level co-occurrence matrix (GLCM). 2008 International Symposium on Telecommunications, 656-659.
16. Vatamanu, O.A., & Ionescu, M. (2012). Data Mining in digital image processing using the Gabor filters algorithm. 2012 IEEE 8th International Conference on Intelligent Computer Communication and Processing, 137-140.
17. Boughida, A., Kouahla, M.N., & Lafifi, Y. (2021). A novel approach for facial expression recognition based on Gabor filters and genetic algorithm. Evolving Systems, 13, 331 - 345.
18. Hajraoui, A., & Sabri, M. (2014). Face Detection Algorithm based on Skin Detection, Watershed Method and Gabor Filters. International Journal of Computer Applications, 94, 33-39.
19. Chuan, L., Xi, Q., Zhu, G., Wei, J., & Lin, C. (2013). Face Detection Algorithm Based on Multi-orientation Gabor Filters and Feature Fusion. Indonesian Journal of Electrical Engineering and Computer Science, 11, 5986-5994.
20. Kumar, S.A., & Sasikala, S. (2021). Disease Detection in Tomato Leaves using Machine Learning and Statistical Feature Fusion. 2021 International Conference on Advancements in Electrical, Electronics, Communication, Computing and Automation (ICAECA), 1-6.
21. Wang, X., Cheng, X., Wu, X., Zhou, H., Chen, X., & Wang, L. (2018). Design of traffic light identification scheme based on TensorFlow and HSV color space. Journal of Physics: Conference Series, 1074.
22. Vijaya Kumar S, Naveen Lazarus M, Nagaraju C (2010). A Novel Method for the Detection of Microcalcifications Based on Multi-scale Morphological Gradient Watershed Segmentation Algorithm. International Journal of Engineering Science and Technology Vol. 2(7), 2616-2622.
23. Rao, G.S., & Srikrishna, A. (2021). Contrast Enhancement of Poor-Quality Satellite Images Through Morphological Operations. Traitement du Signal, 38, 821-827.
24. Czosnek, H. and Laterrot, H. (1997). A worldwide survey of tomato yellow leaf curl viruses. Archives of Virology. 142(7): 1391–1406.
25. Xu, Y., Zhang, S., Shen, J., Wu, Z., Du, Z. and Gao, F. (2021). The phylogeographic history of tomato mosaic virus in Eurasia. Virology. 554: 42–47.
26. Navas-Castillo, J., Sánchez-Campos, S., Díaz, J. A., Sáez-Alonso, E. and Moriones, E. (1999). Tomato Yellow Leaf Curl Virus-Is Causes a Novel Disease of Common Bean and Severe Epidemics in Tomato in Spain. Plant Disease. 83(1): 29–32.
27. https://github.com/spMohanty/PlantVillage-Dataset.

28. Zhao Y, Liu J, Li H and Li G. (2008). Improved watershed algorithm for dowels image segmentation. 7th World Congress on Intelligent Control and Automation, Chongqing, pp. 7644-7648, doi: 10.1109/WCICA.2008.4594115.
29. Li, Q. Q., & Li, W. (2014). An Improved Watershed Segmentation Algorithm for Bridge Image. Applied Mechanics and Materials, 513–517, 3691–3694.
30. Zhang, Y., & Xu, D. Q. (2012). Improved Watershed Algorithm for Cell Image Segmentation. Advanced Materials Research, 546–547, 464–468.
31. Hai B.Z, Xie R.Y and Yuan P.Y. (2016). An Applied Research on Improved Watershed Algorithm in Medical Image Segmentation. International Journal of Signal Processing, Image Processing and Pattern Recognition Vol.9, No.11, pp.191-198.
32. Roshni V.S, Raju G. (2011). Image Segmentation using Multiresolution Texture Gradient and Watershed Algorithm. International Journal of Computer Applications. Volume 22 - Number 6, 21-28.
33. Sebastiani, G., & Stramaglia, S. (1997). A Bayesian approach for the median filter in image processing. Signal Processing, 62(3), 303–309. doi:10.1016/s0165-1684(97)00131-x
34. Liang Y and Gao Y. (2013). A Median Filtering Algorithm Based on Selected Point in Digital Image. International Conference on Information Science and Cloud Computing Companion, Guangzhou, China, 2013, pp. 757-762, doi: 10.1109/ISCC-C.2013.18.
35. Xie, G., Guo, B., Huang, Z., Zheng, Y., & Yan, Y. (2020). Combination of Dominant Color Descriptor and Hu Moments in Consistent Zone for Content Based Image Retrieval. IEEE Access, VOLUME 8, 146284-146299. doi:10.1109/access.2020.3015285