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

Segmentation Algorithms For Accurate Decision Of Banana Leaf Diseases In Precision Agriculture

Vishnu Prabhakar. V[1], Dr. N. Sudha[2]

[1]Research Scholar, Department Of Computer Science, Bishop Appasamycollege Of Arts And Science, Coimbatore, Tamil Nadu.
[2]Associate Professor, Department Of Computer Science, Bishop Appasamycollege Of Arts And Science, Coimbatore, Tamil Nadu.
Email Id: Vishnupchittur@Gmail.Com[1], Sudhanatarajan105@Gmail.Com[2]

Banana cultivation plays a crucial role in the agricultural economy, but its productivity is significantly affected by various leaf diseases. Early and accurate identification of banana leaf diseases is essential for effective disease management and yield optimization. This study presents an automated approach for detecting and classifying banana leaf diseases using a segmentation algorithm. The proposed method involves image preprocessing, segmentation, and feature extraction techniques to isolate diseased regions from healthy leaf areas. By applying advanced image processing and machine learning techniques, the system efficiently identifies common banana leaf diseases such as Sigatoka, Cordana, and bacterial infections. The results demonstrate the effectiveness of the segmentation algorithm in accurately distinguishing diseased areas, thereby assisting farmers and researchers in early disease detection. This approach enhances precision agriculture by providing a reliable and automated solution for monitoring plant health, ultimately contributing to improved crop management and reduced economic losses.

Keywords: Banana leaf disease, Image segmentation, Disease identification, Data Mining, Precision agriculture.

1. INTRODUCTION

Banana is one of the most widely cultivated and consumed fruits worldwide, contributing significantly to food security and the agricultural economy. It serves as a staple food for millions of people, particularly in tropical and subtropical regions, and is also an important cash crop for small-scale and commercial farmers. Countries in Asia, Africa, and Latin America dominate global banana production, with India, Ecuador, Brazil, and the Philippines being the largest producers. The demand for bananas continues to rise due to their high nutritional value, affordability, and year-round availability. However, banana cultivation faces several challenges, with plant diseases being one of the most critical threats affecting production and profitability.

Banana plants are highly susceptible to various fungal, bacterial, and viral infections, which can lead to significant yield losses if not detected and controlled in time. Among these, banana leaf diseases pose a severe threat as they affect the plant's photosynthetic ability, weaken plant growth, and ultimately reduce fruit production. Some of the most common banana leaf diseases include Cordana, Sigatoka and Pestalotiopsis Disease. These diseases spread rapidly, especially in humid tropical environments, and can cause devastating outbreaks if not properly managed. Farmers traditionally rely on manual inspection to identify leaf diseases, which is often inefficient, time-consuming, and prone to human error. Moreover, disease symptoms may appear similar in the early stages, making it difficult for untrained individuals to differentiate between various infections. With the advancement of computer vision and artificial intelligence, automated techniques for plant disease detection have gained increasing attention in recent years. Image processing and segmentation algorithms have proven to be highly effective in identifying infected plant areas, allowing for early diagnosis and timely intervention. Segmentation is a crucial step in image processing that divides an image into meaningful regions, enabling precise identification of diseased and healthy parts of a banana leaf. By applying segmentation techniques, the system can extract critical features from leaf images, such as color, texture, and shape, to classify different types of infections accurately. The main objective of this study is to develop an automated system that utilizes segmentation algorithms for the identification of banana leaf diseases. This approach aims to assist farmers, agronomists, and researchers in detecting diseases at an early stage, thereby improving crop management strategies and reducing economic losses. By integrating image processing and machine learning techniques, the system can analyze leaf images, segment diseased regions, and classify diseases with high accuracy. This automated detection system eliminates the need for constant manual monitoring, making it a cost-effective and efficient solution for banana farmers.

The implementation of segmentation algorithms in disease identification is a significant step towards precision agriculture, where technology is used to optimize farming practices. The adoption of such advanced systems will not only help in reducing the excessive use of pesticides but also contribute to sustainable agricultural practices by promoting targeted disease control measures. Furthermore, by providing a real-time and automated disease detection tool, this study enhances decision-making for farmers, enabling them to take preventive actions before the disease spreads to an uncontrollable level. In conclusion, this research highlights the importance of segmentation algorithms in the identification of banana leaf diseases, emphasizing their role in precision agriculture. By developing an automated disease detection system, farmers can ensure better crop protection, improved yields, and increased profitability. The integration of artificial intelligence and image processing in agriculture is revolutionizing the way diseases are detected and managed, paving the way for a more sustainable and efficient farming future.

2. METHODOLOGY AND ALGORITHMS

The methodology employed in this study for image segmentation involves the application of various advanced segmentation algorithms, including **K-Means**, **Mean Shift**, **Quick Shift**,

Watershed, **Felzenszwalb's**, **SLIC**, **Threshold Segmentation** and **Enhanced Adaptive Thresholding with Morphological Processing (EATMP) segmentation**. Each algorithm was chosen based on its ability to effectively separate regions of interest in the banana leaf images, particularly focusing on diseased and healthy areas. These algorithms were implemented and compared to determine the most efficient and accurate method for identifying banana leaf diseases. In addition to segmentation algorithms, performance evaluation metrics such as **Mean Squared Error (MSE)**, **Peak Signal-to-Noise Ratio (PSNR)**, and **Structural Similarity Index Measure (SSIM)** were used to quantify the quality of segmentation results. MSE measures the average squared difference between the original and segmented images, providing insight into pixel-level discrepancies. PSNR, derived from MSE, evaluates the ratio between the maximum possible pixel value and the noise introduced by segmentation, where higher values indicate better image quality. SSIM, on the other hand, assesses the perceived structural similarity between the images by considering luminance, contrast, and texture, providing a more comprehensive measure of visual quality. By applying these metrics, we were able to objectively compare the effectiveness of each segmentation algorithm in accurately identifying and delineating the diseased regions on banana leaves.

i) K-Means Segmentation

K-Means Segmentation is a popular clustering technique used in image processing to partition an image into distinct regions based on pixel similarity. It is an unsupervised machine learning algorithm that groups similar data points into K clusters by minimizing the variance within each cluster. In image segmentation, K-Means is widely used to separate objects, detect diseased areas in plants, and enhance image analysis tasks.

How K-Means Segmentation Works

1. Choose the Number of Clusters (K): The user defines the number of clusters (K) to divide the image into. For example, if K=3, the algorithm will classify the pixels into three groups (e.g., healthy leaf, diseased region, and background).
2. Initialize Cluster Centroids: The algorithm randomly selects K initial cluster centers (centroids) within the feature space (e.g., pixel color values in RGB or grayscale).
3. Assign Pixels to Clusters: Each pixel in the image is assigned to the nearest centroid based on a similarity measure, such as Euclidean distance. Pixels with similar characteristics (color, intensity, or texture) are grouped together.
4. Update Centroids: After assigning pixels, new centroids are computed by averaging the pixel values in each cluster.
5. Repeat Until Convergence: Steps 3 and 4 are repeated iteratively until centroids remain stable (i.e., no significant changes occur in pixel assignments).

ii) Mean shift Segmentation

Mean Shift Segmentation is a clustering-based segmentation technique used in image processing to partition an image into meaningful regions. Unlike **K-Means**, which requires specifying the number of clusters in advance, Mean Shift is a **density-based** algorithm that automatically determines the number of clusters by identifying high-density regions in the feature space. It is widely used in object detection, image segmentation, and medical image analysis due to its ability to handle complex structures and textures effectively.

How Mean Shift Segmentation Works

1. **Initialize a Window (Bandwidth Selection):** The algorithm starts by placing a window (kernel) at a randomly selected pixel in the feature space (e.g., color or intensity values). The window size, called **bandwidth (h)**, determines how far the algorithm searches for high-density regions.
2. **Compute the Mean (Centroid Shift):** Within the window, the algorithm calculates the mean of all pixel values (color, intensity, or spatial features). The window is then shifted toward the mean, moving towards a denser region of data points.
3. **Iterate Until Convergence:** The shifting process continues iteratively until the window stabilizes at a point where further movement is minimal. This stabilized point is considered the **mode** (local peak) of the density function, representing a distinct segment.
4. **Assign Pixels to Segments:** After convergence, pixels that belong to the same mode are grouped together, forming **segmented regions** in the image.

iii) **Quick Shift Segmentation**

Quick Shift Segmentation is a fast and efficient image segmentation technique that is commonly used in computer vision tasks. It is a **mode-seeking algorithm** that clusters pixels based on intensity and spatial proximity, making it useful for segmenting objects in an image. Unlike **K-Means** and **Mean Shift**, Quick Shift works by building a hierarchical tree structure of pixels based on local density estimates, making it effective for segmenting complex structures.

How Quick Shift Segmentation Works

1. **Compute Pixel Similarity:** The algorithm calculates a distance metric based on color (e.g., RGB values) and spatial coordinates (x, y).
2. **Identify High-Density Pixels:** It searches for **local maxima** in pixel densities and groups pixels based on the shortest path to their nearest high-density neighbor.
3. **Construct a Tree Structure:** A **tree-based representation** of pixels is built, where each pixel is connected to a parent based on similarity and spatial proximity.
4. **Segment Image into Clusters:** Pixels that are closely linked in the tree are grouped together to form **segmented regions**.

iv) **Watershed Segmentation**

Watershed Segmentation is a powerful **region-based segmentation** technique used in image processing to separate objects in an image, especially when objects are touching or overlapping. The method is inspired by the concept of **topography**—it treats an image as a **terrain** where intensity values represent elevations, and segmentation works like **water flowing into valleys**. This technique is widely used in medical imaging, agricultural analysis, and object recognition tasks.

How Watershed Segmentation Works

1. **Convert Image to Grayscale**: If the image is in RGB, it is converted into a grayscale image for processing.
2. **Compute Gradient Magnitude**: The gradient (edge information) of the image is calculated to highlight object boundaries.
3. **Apply Marker-Based Segmentation**: Two sets of markers are created: **Foreground markers** (representing objects of interest).,**Background markers** (representing non-interest areas).
4. **Perform Watershed Algorithm:** The algorithm treats the image like a **topographic map** where high gradients act as ridges (object boundaries). Water is simulated to flow from high elevations into valleys (regions with low intensity). As water fills the regions, it merges only when it reaches a ridge, effectively separating the objects.
5. **Segmentation Output :** The segmented objects are labeled, and boundaries are clearly defined.

v) **Felzenswalb's Segmentation**

Felzenszwalb's Segmentation is an efficient and widely used **graph-based image segmentation** technique. It is designed to generate visually meaningful segments by grouping pixels based on similarity in **color, texture, and spatial proximity**. Unlike K-Means or Watershed, which require predefined parameters, Felzenszwalb's algorithm dynamically adjusts segment sizes based on the input image, making it ideal for **natural image processing and object detection**.

How Felzenszwalb's Segmentation Works

1. **Graph Construction:** The algorithm treats an image as a **graph**, where each pixel is a **node** and edges connect neighboring pixels.
2. **Compute Pixel Similarity:** Each edge is assigned a **weight** based on the difference between pixel intensities (color, brightness, texture).
3. **Merge Similar Regions:** The algorithm starts grouping pixels into segments, ensuring that the internal **similarity** within a segment is **higher** than the difference between two neighboring segments.

4. **Dynamic Region Formation:** The size of segments is **automatically determined** based on an internal threshold.

5. **Final Segmentation Output:** The image is divided into segments, with each segment representing a distinct region based on pixel similarities.

vi) SLIC Segmentation

SLIC (Simple Linear Iterative Clustering) Segmentation is a superpixel-based segmentation technique used in image processing. Unlike traditional segmentation methods, which operate at the pixel level, SLIC groups pixels into small, perceptually meaningful clusters called superpixels. These superpixels help in reducing computational complexity while preserving important image details. SLIC is widely used in medical imaging, object recognition, and agricultural applications, including banana leaf disease detection, where it efficiently segments diseased regions while maintaining clear boundaries.

How SLIC Segmentation Works

1. Convert Image into Lab Color Space: The image is transformed from RGB to the Lab color space, which is more perceptually uniform.

2. Initialize Superpixel Centers: The image is divided into a grid, and initial cluster centers are placed at equally spaced intervals.

3. Assign Pixels to Nearest Superpixels: Each pixel is assigned to the nearest cluster center based on color and spatial proximity.

4. Iterative Refinement (Update Cluster Centers): Cluster centers are updated by averaging the pixel values assigned to them. This step is repeated until convergence, ensuring compact and uniform superpixels.

5. Enforce Connectivity: Small or disjointed clusters are merged to maintain segment continuity.

vii) Threshold Segmentation

Threshold Segmentation is one of the simplest and most widely used image segmentation techniques. It works by converting a grayscale or color image into a binary (black and white) image by applying a threshold value. Pixels with intensity values above the threshold are classified as one group (e.g., foreground), while those below the threshold are classified as another group (e.g., background). This technique is particularly useful in object detection, defect inspection, and medical imaging. In banana leaf disease detection, threshold segmentation helps in isolating diseased regions based on color intensity differences.

How Threshold Segmentation Works

1. **Convert Image to Grayscale:** If the input is an RGB image, it is converted into a grayscale image, as thresholding works best on single-channel images.
2. **Choose a Threshold Value (T):** A predefined or automatically calculated threshold value is selected. Example: If **T = 127** (range 0-255), all pixels **above 127** become **white (255)** and those **below 127** become **black (0)**.
3. **Apply Thresholding Rule :** Each pixel in the image is compared to the threshold **T**:

   If pixel intensity ≥ T → Assign White (Foreground), If pixel intensity < T → Assign Black (Background)

4. **Generate Binary Image:** The output is a segmented image where objects of interest are highlighted while the background is removed.

viii) Enhanced Adaptive Thresholding with Morphological Processing (EATMP) segmentation

The **EATMP** algorithm enhances traditional thresholding by dynamically adjusting the threshold based on local image characteristics. It begins by converting the input image to grayscale and applying **Gaussian smoothing** to reduce noise. A **local adaptive threshold** is computed using the **mean and standard deviation** of pixel intensities within a neighborhood window. The threshold is fine-tuned using a sensitivity factor (**k**) to prevent over-segmentation. Pixels with intensities below the computed threshold are classified as foreground, while others are set as background. To further refine the segmented output, **median filtering** removes small noise artifacts, and **morphological operations (closing and opening)** enhance the object boundaries while eliminating unnecessary speckles.

How EATMP works

1. **Preprocessing:** Convert the input image to **grayscale** and apply **Gaussian smoothing** to reduce noise while preserving important features.

2. Adaptive Threshold Calculation: For each pixel (x,y) compute the **local mean** μ(x,y) and **standard deviation** σ(x,y) within a neighborhood window. Determine the adaptive threshold using:

T(x,y)=μ(x,y)+k·σ(x,y) where **kkk** is a sensitivity factor (e.g., **0.15**). Pixels **below** T(x,y)T(x,y)T(x,y) are classified as **foreground (diseased regions)**.

3. Noise Reduction & Refinement: Apply **median filtering** to remove small noise spots and enhance segmentation clarity.

4. Morphological Processing & Output: Use **closing** (to fill small gaps) and **opening** (to remove tiny speckles) with a **disk-shaped structuring element**. Finally, display the **segmented image**, highlighting diseased areas with minimal background interference.

4. DATASET

The original images are considered for the evaluation. There are three different types of diseased banana leaves taken for this, they are namely Cordana, Sigatoka, and Pestalotiopsis.

Input images:

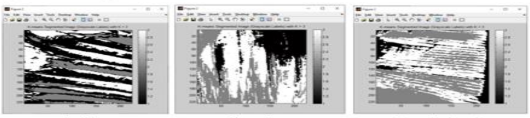
Cordana Sigatoka Pestalotiopsis

Input Images

Fig (1)

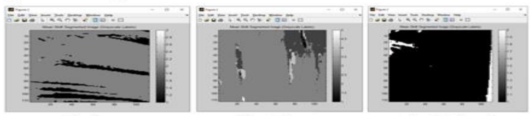
Segmentation Results:

(1) K-Means Segmentation



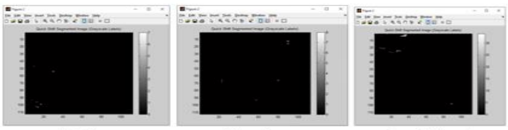
Cordana Sigatoka Pestalotiopsis

(2) Mean shift Segmentation



Cordana Sigatoka Pestalotiopsis

(3) Quick shift Segmentation

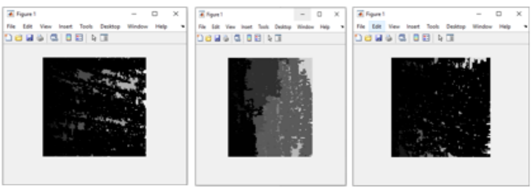


Cordana Sigatoka Pestalotiopsis

(4) Watershed Segmentation

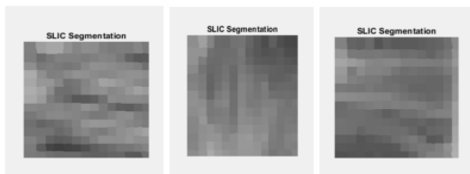
Cordana Sigatoka Pestalotiopsis

(5) Felzenswalb's Segmentation



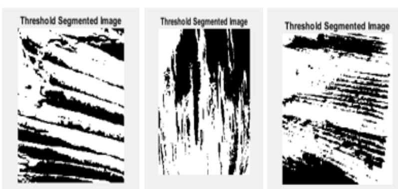
Cordana Sigatoka Pestalotiopsis

(6) SLIC Segmentation



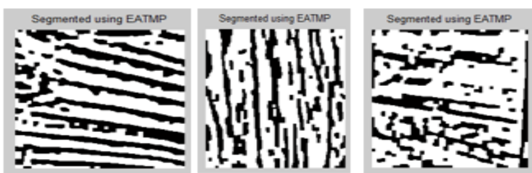
Cordana Sigatoka Pestalotiopsis

(7) Threshold segmentation



Cordana Sigatoka Pestalotiopsis

(8) EATMP Segmentation



Cordana Sigatoka Pestalotiopsis

Fig(1) shows the original image of Banana leaf disease such as Cordana, Sigatoka, Pestalotopsis Fig(2) depicts the results after segmentation with the algorithms such as the K-means Segmentation, Mean shift algorithm, Quick shift segmentation, Watershed segmentation, Felzenswalb's Segmentation, SLIC Segmentation, Threshold Segmentation and Enhanced Adaptive Thresholding with Morphological Processing (EATMP) segmentation.

5. EVALUATION METRICS

MSE

MSE (Mean-Squared Error) offers a straightforward and efficient method for assessing differences in images, but it has its drawbacks. It considers all pixel errors the same, neglecting how humans perceive images, where some details are more important than others. Furthermore, MSE can be overly affected by noise or minor, high-frequency changes, which may lead to an inaccurate representation of the image's overall quality.

PSNR

A crucial metric in image processing for assessing how well a reconstructed image compares to the original is the Peak Signal-to-Noise Ratio (PSNR). In essence, it measures the amount of noise that has been added to the image as a result of compression or transmission. The mean squared error (MSE) between the original and reconstructed pictures is compared to determine PSNR. The average squared difference between the corresponding pixel values in the two images is known as the MSE. A higher PSNR value and a lower MSE signify a tighter likeness between the images. Better image quality is indicated by greater PSNR values, which are commonly reported in decibels (dB).

SSIM

SSIM (Structural Similarity Index Measure) is a perceptual metric used to assess the visual quality of images. Unlike traditional metrics like Mean Squared Error (MSE) or Peak Signal-to-Noise Ratio (PSNR), which only measure pixel-wise differences, SSIM takes into account structural information such as luminance, contrast, and texture. It evaluates the similarity between two images based on how similar their structural patterns are, rather than just raw intensity differences. SSIM values range from 0 to 1, where 1 indicates perfect structural similarity and 0 indicates no similarity. It is widely used in image compression, restoration, and segmentation tasks to evaluate the quality of segmented images or reconstructed images after processing.

	MSE							
	K-Means Segmentation	Mean shift Segmentation	Quick shift Segmentation	Watershed Segmentation	Felzenswalb's Segmentation	SLIC Segmentation	Threshold Segmentation	EATMP segmentation
Cordana	70.305	72.76	97.75	61.51	87.89	67.50	46.29	44.12

Sigatoka	69.522	68.98	97.05	60.93	73.38	67.52	39.42	37.25
Pestalotiopsis	69.19	96.40	93.14	60.25	90.74	67.23	50.38	47.67

	PSNR							
	K-Means Segmentation	Mean shift Segmentation	Quick shift Segmentation	Watershed Segmentation	Felzenswalb's Segmentation	SLIC Segmentation	Threshold segmentation	EATMP segmentation
Cordana	26.69	29.55	28.26	30.28	28.73	29.87	31.51	33.16
Sigatoka	29.74	29.78	28.29	30.31	29.51	29.87	32.21	34.65
Pestalotiopsis	29.76	28.32	28.47	30.37	28.59	29.89	31.14	33.48

	SSIM							
	K-Means Segmentation	Mean shift Segmentation	Quick shift Segmentation	Watershed Segmentation	Felzenswalb's Segmentation	SLIC Segmentation	Threshold Segmentation	EATMP segmentation
Cordana	0.75	0.89	0.96	0.81	0.96	0.8	0.98	0.99
Sigatoka	0.68	0.77	0.94	0.813	0.94	0.93	0.97	0.99
Pestalotiopsis	0.683	0.95	0.97	0.80	0.92	0.94	0.97	0.98

Experimental result shows that the **Enhanced Adaptive Thresholding with Morphological Processing (EATMP)** segmentation gives accurate results for Cordana, Sigatoka and Pestalotiopsis disease when compared to other algorithms.

6. CONCLUSION

In this research work, diseases of banana leaves are identified. The different image processing techniques and algorithms utilized to detect banana leaf diseases were also taken into consideration in this work. The diagnosis of diseased leaves can be effectively accomplished with a variety of segmentation methods. According to the evaluation's findings, the Enhanced Adaptive Thresholding with Morphological processing (EATMP) segmentation outperforms the K-Means, Mean shift, Quick shift, Watershed, Felzenswalb's, Threshold Segmentation and SLIC segmentations in terms of effectiveness. This would lessen the impact of plant diseases and help the agriculture sector become more productive. Additionally, it helps farmers identify diseases in plants early on. This will serve as a proactive step in the process of finding a cure for banana leaf diseases.

REFERENCES

1. Krishnan, V. Gokula, et al. "An automated segmentation and classification model for banana leaf disease detection." Journal of Applied Biology and Biotechnology 10.1 (2022): 213-220.
2. Banerjee, Deepak, et al. "Precision agriculture: classifying banana leaf diseases with hybrid deep learning models." 2023 IEEE 8th International Conference for Convergence in Technology (I2CT). IEEE, 2023.
3. Chaudhari, Vandana and Manoj P. Patil. “Banana leaf disease detection using K-means clustering and Feature extraction techniques.” 2020 International Conference on Advances in Computing, Communication & Materials (ICACCM) (2020): 126-130.
4. Patil, Preet et al. “Banana Plant Disease Detection using Image Processing.” 2024 IEEE 9th International Conference for Convergence in Technology (I2CT) (2024): 1-5.
5. Vijai Singh a, A.K. Misra b. Detection of plant leaf diseases using image segmentation and soft computing techniques. 2016 INFORMATION PROCESSING IN AGRICULTURE 4 (2017) 41–49
6. Anusha Rao and S.B. Kulkarni. A Hybrid Approach for Plant Leaf Disease Detection and Classification Using Digital Image Processing Methods. 2020 International Journal of Electrical Engineering & Education.
7. Ms. Kiran R. Gavhale, Prof. Ujwalla Gawande. An Overview of the Research on Plant Leaves Disease detection using Image Processing Techniques. 2014 IOSR Journal of Computer Engineering (IOSR-JCE).
8. Iqbaldeep Kaur, Gifty Aggarwal and Amit Verma. Detection and Classification of Disease Affected Region of Plant Leaves using Image Processing Technique. 2016 Indian Journal of Science and Technology.
9. Chaitali G. Dhaware, Mrs. K.H. Wanjale. A Modern Approach for Plant Leaf Disease