

GTADAD: A Novel Grayscale Thresholding Algorithm for Fruit Disease Area Detection

OPEN ACCESS

Volume: 13

Special Issue: 3

Month: February

Year: 2026

P-ISSN: 2321-788X

E-ISSN: 2582-0397

Citation:

Henila, M. "GTADAD: A Novel Grayscale Thresholding Algorithm for Fruit Disease Area Detection." *Shanlax International Journal of Arts, Science and Humanities*, vol. 13, no. 3, 2026, pp. 164–70.

DOI:

<https://doi.org/10.34293/sijash.v13iS3-i1-Feb.10276>

Dr. M. Henila

*Associate Professor, Department of BCA, Ethiraj College for Women
Chennai, Tamil Nadu, India*

Abstract

Grayscale thresholding algorithms are vital in computer vision for simplifying images into binary (black and white) data, which reduces computational complexity and isolates objects of interest from the background. Grayscale thresholding algorithms segment images by converting pixels into black or white based on intensity. They are essential for object detection and image analysis. Identifying the area of interest that is diseased or defective on fruit is another important area of agricultural research, particularly when sorting them using fruit sorters as fresh or infected fruit. A novel algorithm, GTADAD [Grayscale Thresholding Algorithm for diseased Area Detection], has been proposed in this paper for identifying the diseased area on the fruit image. The proposed algorithm is a simple thresholding algorithm that uses a single, user-defined value to classify all pixels. If $\text{pixel} > T$, it becomes white; otherwise, it becomes black. Each gray pixel is converted to a new pixel value using a data transformation technique that uses the $\log 10$ function. The square root of the sum of all converted pixel values is the 'T' value for identifying and extracting the diseased or defective area from the fruit image. 150 fruit images [orange, banana, apple, papaya, and mango] were collected from Kaggle, Medley, and Google. Output obtained using the proposed GTADAD algorithm was compared with other famous grayscale algorithms like the Triangle method, Sauvola's method, and Niblack's method. Keeping the universal grayscale OTSU's output image as the ground truth image, GTADAD, the Triangle method, Sauvola's method, and Niblack's method outputs were compared. The result showed that the proposed GTADAD algorithm outperformed the other three algorithms both visually and statistically. Metrics such as the Jaccard index, Dice coefficient, mean squared error, and structural similarity index demonstrated that the proposed algorithm, GTADAD, outperformed in the metrics analysis and achieved a good accuracy rate in segmenting the diseased area.

Keywords: Gray-Scale Threshold, GTADAD, Fruit Disease, Triangle Method, Sauvola Method, Niblack Method

Introduction

Grayscale thresholding algorithms play a foundational role in computer vision for fruit disease identification by enabling image segmentation, which isolates the region of interest (infected areas) from the healthy fruit skin and background. By converting color images into grayscale and applying intensity-based thresholds, these algorithms simplify complex visual data into binary images (black or white), allowing for fast, automated, and accurate detection of lesions, spots, and rot.

Thresholding is used to separate diseased parts— which often appear as darker, lower-intensity regions—from the healthy, lighter-colored skin of the fruit. These algorithms help in converting RGB images to grayscale, which is a crucial, high-speed step in preprocessing to remove irrelevant background noise and focus on crucial pixel intensity data. Once segmented, the binary images are used to compute geometric and morphological features (e.g., area, perimeter) and to isolate diseased spots for extracting color and texture features (using Gray-Level Co-occurrence Matrix, GLCM) to identify the specific type of disease. Compared to more complex segmentation methods, grayscale thresholding algorithms (like Otsu’s) provide rapid, real-time results, which are essential for industrial-scale, automated fruit sorting machines. So, a novel grayscale thresholding algorithm for fruit disease area detection, namely GTADAD, has been proposed in this paper.

Literature Survey

A few literature works related to gray-scale thresholding and fruit disease identification are discussed below:

Image thresholding has long been recognized as a fundamental technique in digital image processing, with applications ranging from medical imaging to agricultural analysis. In the context of fruit disease detection, several approaches have been proposed to improve segmentation accuracy and efficiency. Early work [1] introduced a global thresholding method based on object size, where thresholds are selected to maximize recall and precision within particularly in terms of PSNR, SSIM, and FSIM. A comprehensive review of thresholding strategies highlighted [5] the diversity of available techniques, ranging from classical to advanced, and emphasized the importance of selecting methods based on image characteristics. Practical implementations, including Python code for binary image ROI extraction, further underscored the accessibility of these approaches. Other notable contributions [6] include the codebook-based multidirectional thresholding algorithm, which iteratively searches foreground and background pixels using directional standard deviations, achieving high segmentation accuracy as measured by misclassification error and Jaccard index. Similarly, a variance-based local thresholding method [10] analyzed gradient orientation histograms by clustering pixels into subsets with varying angular intervals. This approach effectively captured pixel correlations, maintained stable performance with increasing thresholds, and demonstrated competitive processing times suitable for real-time applications.

Although these methods demonstrate strong segmentation capabilities, challenges remain in precisely isolating diseased regions in fruit images. Defined size intervals. This approach, implemented in tools such as ImageJ and CellProfiler, relies on component tree representation for efficient computation. Subsequent studies [2] developed the Gray Scale Thresholding Method (GSTM), which converts color images into grayscale and determines thresholds using pixel intensity values. When combined with deep convolutional neural networks, GSTM has been successfully applied to automated apple sorting systems. Building on Otsu’s method, researchers[3] proposed an iterative trichotomous thresholding technique. This method partitions images into three classes instead of two, refining segmentation iteratively until convergence. Morphological filtering is then applied to reduce noise and improve the accuracy of diseased region extraction. Optimization-based strategies have also been explored [4]. The Modified Gray Wolf Optimizer (MGWO) enhances multi-level thresholding by improving leader selection and mutation strategies. Using Otsu and Kapur entropy as objective functions, MGWO has demonstrated superior performance on benchmark datasets,

Many existing approaches either emphasize general image segmentation or focus on optimization without tailoring specifically to fruit disease detection. Noise sensitivity, variability across fruit types, and computational efficiency are recurring issues.

To address these limitations, this paper introduces GTADAD (Gray-Scale Thresholding Algorithm for Diseased Area Detection). The algorithm is designed to:

- Enhance segmentation accuracy specifically for diseased fruit regions.
- Reduce false detections and background noise.
- Maintain efficiency for real-time fruit sorting systems.
- Provide a robust framework adaptable across different fruit varieties.

Materials and Methods

To develop and test the efficiency of the proposed GTADAD algorithm, it was coded using Python Language using Spider Software. A total of 150 sample images of 5 different fruit varieties were taken. Images were sourced from from Kaggle, Medley, and Google.

Table 1 Image Dataset

S.No.	Fruit Type	Number of Samples with Disease
1	Apple	34
2	Banana	29
3	Gauva	16
4	Papaya	36
5	Mango	35
	Total	150

Table 1 shows the Number of sample fruit images taken for experimentation. All images were sourced from the internet. The pie chart in Fig.1 shows the percentage of fruit images taken for experimentation.

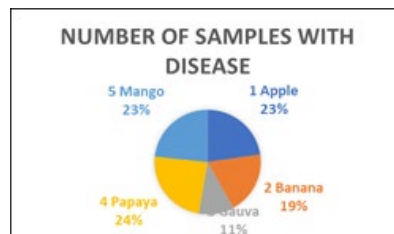


Fig. 1 Percentage of samples taken for research

Proposed Gtadad Algorithm

In the proposed algorithm 1: GTADAD, the given RGB image is converted to a gray-scale image. Later, using the data transformation technique \log_{10} of each pixel was calculated and summed. To normalize the value, the square root of the sum was calculated. The threshold value is then calculated by adding a constant value of 75. The constant value is added to fine-tune segmentation; ensuring small variations in lighting or noise do not result in false foreground detection. It acts as a bias to refine the threshold value, allowing better separation between foreground objects and background noise. And finally, Eqn. (1) is used to segment the image into background and foreground images

$$S(x,y) = \begin{cases} 1 & \text{if } gri(x,y) > Th \\ 0 & \text{if } gri(x,y) \geq Th \end{cases} \quad (1)$$

where, $S(x,y)$ is the binarized image, $gri(x,y)$ is the segmented image got using the proposed GTADAD algorithm, and 'Th' is the threshold value obtained as output from GTADAD.

Algorithm 1: GTADAD – Proposed Thresholding

```

Input : Color image
Output: Binary segmented image
Step 1: Acquire the RGB color image (imag)
Step 2: gri = RGBtoGRAY(imag), sum = 0
Step 3: for I = 1 to 255
        for j = 1 to 255
            sum = sum + log10(gri[I,j])
Step 4: Th = Sqrt(sum)+75
Step 5: Binarize the image using Eqn.(1), to get S
Step 6: Print S
    
```

Gray-Scale thresholding methods used for Comparison

Otsu [7]

Otsu’s method is a widely used algorithm for automatic image thresholding. It determines the optimal threshold by analyzing the image histogram, which represents the distribution of pixel intensities. The goal is to separate the pixels into two distinct classes—foreground and background—such that the **intra-class variance** (variance within each class) is minimized, or equivalently, the **inter-class variance** (variance between classes) is maximized. This ensures that the chosen threshold provides the clearest possible distinction between the two regions.

Sauvola [8]

Sauvola’s algorithm builds upon Niblack’s local thresholding approach by introducing an adaptive factor that adjusts the threshold according to both the local mean intensity and the contrast within a neighborhood. This dynamic adjustment makes the method more effective in handling images with uneven illumination or varying contrast levels. As a result, Sauvola’s method produces more reliable binarization outcomes, particularly in complex regions where traditional fixed-factor methods may fail.

Triangle Algorithm

The Triangle algorithm is a global thresholding technique that determines the threshold directly from the image histogram. It constructs a right triangle between the histogram’s highest peak and the farthest end of the intensity distribution. The threshold is selected at the point where the distance between the hypotenuse of the triangle and the histogram curve is greatest. This method is especially useful for images characterized by a strong background peak and a weaker foreground peak, making it suitable for cases where object regions are less prominent compared to the background.

Niblack Algorithm[9]

Niblack’s method is a local thresholding technique that calculates a threshold for each pixel based on the statistical properties of its surrounding neighborhood. The threshold is defined as:

$$t = m + (k \times \text{std}) \quad (2)$$

where t is the threshold value, m is the mean intensity of the neighborhood, “std” is the standard deviation of pixel intensities, and k is a constant that controls the influence of the standard deviation.

By incorporating both mean and variance, Niblack's method adapts to local variations in intensity, making it effective for segmenting images with textured or non-uniform backgrounds.

Metrics Used for Comparison

MSE (Mean Square Error)

It is computed by averaging the squared intensity of the original input image and the output image pixels as follows;

$$MSE = \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} (e(i, j) * e(i, j)) \quad (3)$$

where m = no. of rows and n = no. of columns.

In Eqn.(3), $e(i,j)$ is the error difference between the original and the proposed output image.

Dice Co-efficient

The Dice coefficient or Sorensen-Dice index (Eqn.(4)), is a key metric in image segmentation that measures the similarity between a predicted segmentation mask and the ground truth by calculating the overlap. Ranging from 0 (no overlap) to 1 (perfect match).

$$\text{Dice} = 2 * |X \cap Y| / (|X| + |Y|) \quad (4)$$

where X and Y are two sets. A set with vertical bars either side refers to the number of elements in that pixel set, $|X|$ means the number of elements in set

$X \cap Y$ is used to represent the intersection of two pixel sets, and means the elements that are common to both sets.

Jaccard Index

The Jaccard index, often referred to as Intersection-over-Union (IoU), is another measure of similarity between two sets of pixels. It compares the size of the intersection of the predicted and ground truth regions with the size of their union. Like the Dice coefficient, its values range from 0 to 1, with higher values indicating stronger similarity. The Jaccard index is widely used in evaluating segmentation accuracy and object detection tasks.

SSIM (Structural Similarity Index Measure)

The Structural Similarity Index Measure is a perceptual metric designed to assess image quality by comparing structural information between two images. Unlike MSE, which only considers pixel-wise differences, SSIM evaluates luminance, contrast, and structural similarity. It provides a score between -1 and 1 , where 1 indicates perfect similarity. SSIM is often used in applications such as image compression, restoration, and quality assessment, as it aligns more closely with human visual perception.

Results and Discussions

All algorithms were coded using the Python Language. Outputs with segmented images and corresponding metrics (keeping the Otsu image as the ground truth image) were generated for 150 sample images. Out of the five sample images, the output is represented in Fig. 2.

Fig.2.(a),(b),(c),(d),(e) show the input gray image (RGB image converted to gray-scale image) and segmented images obtained as output using OTSU, Niblack, Sauvola, Triangle, and the proposed GTADAD algorithms. It can be observed that the visual accuracy is higher in segmenting the diseased area using GTADAD than other thresholding algorithms. Comparison among all algorithms (Table 2) in segmenting the area of interest was done using metrics like MSE, Jaccard Index, SSIM, and PSNR, keeping OTSU's output as the ground truth image.

From Table 2, it is evident that the similarity matrices, such as Jaccard index and SSIM, showed higher similarity in comparing the proposed GTADAD with OTSU than other algorithms. Also, PSNR is lower, and MSE is lower when GTADAD's output image was compared with other images improving interpretability and further supporting real-time fruit sorting applications.

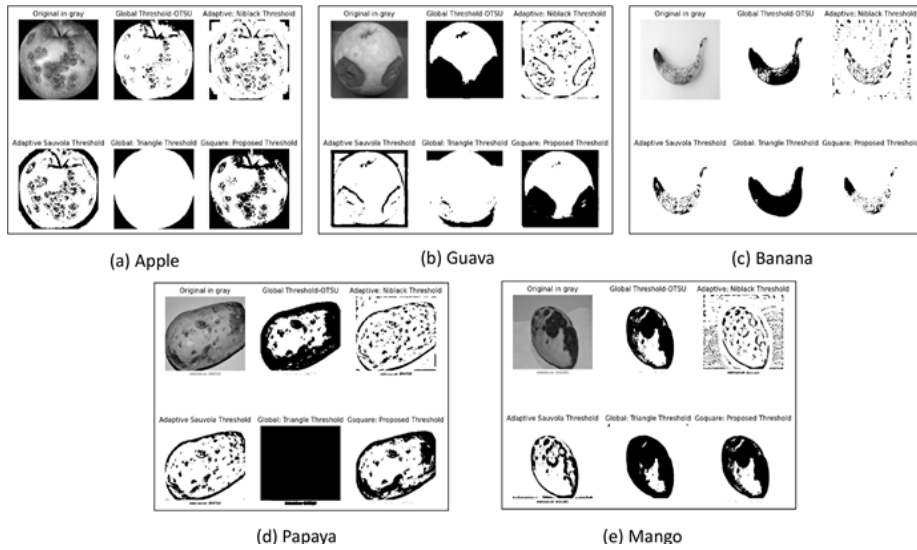


Fig 2: Visual Comparison of the outputs obtained by OTSU, Sauvola, Niblack, Triangle, and Proposed GTADAD

Table 2: Metrics-based comparison keeping Otsu image as the Ground Truth Image






S.No.	Color Image	Metrics	Otsu Vs Proposed GTADAD	Otsu Vs Niblack	Otsu Vs Sauvola	Otsu Vs Triangle
1		Jaccard Index (IoU)	0.9808	0.7205	0.8441	0.9200
		Dice Coefficient	0.9903	0.8376	0.9155	0.9583
		Mean Squared Error (MSE)	0.0144	0.2615	0.1360	0.0601
		Structural Similarity Index (SSIM)	0.9314	0.3428	0.6884	0.8183
2		Jaccard Index (IoU)	0.8108	0.6112	0.6735	0.0878
		Dice Coefficient:	0.8955	0.7587	0.8049	0.1615
		Mean Squared Error (MSE)	0.1329	0.3421	0.2732	0.5195
		Structural Similarity Index (SSIM)	0.6074	0.3873	0.5005	0.2916
3		Jaccard Index (IoU):	0.9006	0.8667	0.9095	0.9181
		Dice Coefficient:	0.9477	0.9286	0.9526	0.9573
		Mean Squared Error (MSE)	0.0958	0.1285	0.0864	0.0711
		Structural Similarity Index (SSIM)	0.8193	0.6660	0.8382	0.8232
4		Jaccard Index (IoU)	0.9051	0.4743	0.5659	0.5259
		Dice Coefficient	0.9502	0.6434	0.7228	0.6893
		Mean Squared Error (MSE)	0.0478	0.4826	0.3492	0.4114
		Structural Similarity Index (SSIM)	0.8615	0.2937	0.5150	0.4550
5		Jaccard Index (IoU)	0.7983	0.7154	0.7757	0.6853
		Dice Coefficient	0.8878	0.8341	0.8737	0.1210
		Mean Squared Error (MSE)	0.1404	0.2580	0.1829	0.2195
		Structural Similarity Index (SSIM)	0.5796	0.4741	0.5539	0.3259

Fig. 3: Bar Diagram Representation - Metrics comparison for Mango Fruit

In Fig. 3, the output obtained for the mango fruit image is depicted using a bar diagram. To prove that the proposed GTADAD outperforms all other compared thresholding algorithms, in the above diagram it can be seen that the Jaccard Index, Dice Co-efficient, and SSIM of GTADAD (bars in blue) values are high and MSE value is the lowest among others. Thus, it is evident that GTADAD can be recommended for identifying and segmenting diseased fruit areas.

Conclusion

The proposed GTADAD algorithm demonstrated superior performance compared to existing gray-scale thresholding techniques. Its effectiveness was evident both in terms of visual accuracy, where diseased regions on fruit surfaces were clearly identified, and through similarity metrics such as the Jaccard Index, which confirmed its robustness in segmentation tasks. The results indicate that GTADAD can be implemented within automated fruit sorting systems to reliably detect and segment diseased areas. This capability enables a practical decision process: fruits identified as healthy can be directed to the market, while those with diseased regions can be discarded, thereby improving quality control in agricultural supply chains. Looking ahead, future work may focus on extending GTADAD to produce color-segmented outputs rather than binary images. Such an enhancement would provide richer visual information, potentially

References

1. Ranefall, Petter, and Carolina Wählby. "Global gray-level thresholding based on object size." *Cytometry Part A* 89.4 (2016): 385-390.
2. Henila, M. "Diseased Area Segmentation Using a Novel Gray-Scale Thresholding Algorithm and Classification Using a New Deep CNN Model for Apple Fruit Sorting." (2025).
3. Priya, MR Vishnu, and M. E. K. Kavitha. "An effective segmentation on gray scale images using iterative triclass Otsu thresholding." *Int. J. Eng. Res. Tech* 3.16 (2015): 1-5.
4. Hu, Pei, Yibo Han, and Zheng Zhang. "Multi- Level Thresholding Color Image Segmentation Using Modified Gray Wolf Optimizer." *Biomimetics* 9.11 (2024): 700.
5. Mohammed, M. Riyaz, and M. Sabibullah. "Thresholding Techniques in Computer Vision Applications." *i-manager's Journal on Image Processing* 11.2 (2024): 27.
6. Reka, R. "Directional Thresholding Algorithm For Gray Scale Image Segmentation". *International Journal of Advances in Signal and Image Sciences* 4.1 (2018): 23-29
7. Nyo, Myat Thet, et al. "Otsu's thresholding technique for MRI image brain tumor segmentation." *Multimedia tools and applications* 81.30 (2022): 43837-43849.
8. Sun, Zichen, et al. "Low-Light Object Detection via Gray-Level Difference Statistics Texture Enhancement and Sauvola Local Thresholding." *2025 International Conference on Aerospace Information Perception and Intelligent Processing (AIPIP)*. IEEE, 2025.
9. Samorodova, O. A., and A. V. Samorodov. "Fast implementation of the Niblack binarization algorithm for microscope image segmentation." *Pattern recognition and image analysis* 26.3 (2016): 548-551.
10. Dong, Lijie, et al. "A Local Thresholding Algorithm for Image Segmentation by Using Gradient Orientation Histogram." *Applied Sciences* 15.17 (2025): 9808.