Scilab case study project on

The Comparative Performance Study of Image Segmentation Using GUI

Deepan Nagarajan

Government Polytechnic College, Srirangam, Tiruchirappalli, Tamil Nadu.

Image Processing

10-07-25

Abstract

Image segmentation is a fundamental technique in computer vision, with applications ranging from medical imaging to object detection. It involves partitioning an image into regions where pixels within each region share similar attributes such as color, intensity, or texture. This study presents a comparative performance analysis of two widely used segmentation techniques: K-means clustering (an unsupervised partition-based method) and thresholding-based methods (including global and adaptive approaches), implemented through an interactive Scilab graphical user interface (GUI) and image processing and computer vision (IPCV) toolbox. The GUI interface enables real-time experimentation with critical parameters (e.g., cluster count K), facilitating a deeper understanding of trade-offs between method complexity and output quality. Results demonstrate that K-means excels in multi-region segmentation but is computationally intensive, while thresholding offers faster execution but struggles with intensity overlap. This work bridges theoretical principles with practical implementation, serving as an educational resource for students and researchers in image processing.

Keywords: Image segmentation, K-Means clustering, Thresholding, Image Processing and Computer Vision (IPCV), Graphical User Interface (GUI).

1 Introduction

Image segmentation is a fundamental technique in image processing, serving as a critical preprocessing step in image analysis, computer vision, and pattern recognition. It involves partitioning a digital image into distinct regions (sets of pixels) to identify objects and boundaries, such as edges or curves. The output of segmentation can be either a collection of segments covering the entire image or a set of extracted contours (resulting from edge detection). Pixels within the same region share similar attributes such as color, intensity, or texture, while adjacent regions exhibit significant differences in these characteristics. The quality of segmentation directly impacts the performance of higher-level computer vision tasks such as object tracking, facial recognition, and autonomous driving.

Traditional segmentation methods can be broadly categorized into:

- **Clustering-based approaches** (e.g., K-Means)
- **Edge detection techniques** (e.g., Sobel, Canny)
- **Region-based methods** (e.g., watershed)
- **Thresholding techniques** (e.g., Otsu's method)

Among these, **K-Means clustering** and **thresholding** are widely adopted due to their simplicity and effectiveness. However, each method has distinct strengths and weaknesses, making them suitable for different scenarios. This study aims to compare their performance and provide practical recommendations for real-world applications. This case study focused on k-means clustering, thresholding, and their procedures.

2 Problem Formulation

Despite advancements in deep learning-based segmentation (e.g., U-Net, Mask R-CNN), traditional methods like K-Means and thresholding remain relevant due to their low computational cost and interpretability. The primary objectives of this study are:

- To implement and analyse K-Means and thresholding for image segmentation.
- To compare their performance using quantitative metrics and qualitative assessment.
- Guidelines for selecting segmentation methods based on image properties.
- To discuss practical insights for improving segmentation outcomes.

3 Basic concepts related to the topic

3.1 Image segmentation based on K- means clustering

K-Means Clustering is an unsupervised machine learning algorithm that partitions an unlabeled dataset into distinct groups, or clusters, based on similarity. The algorithm operates as follows:

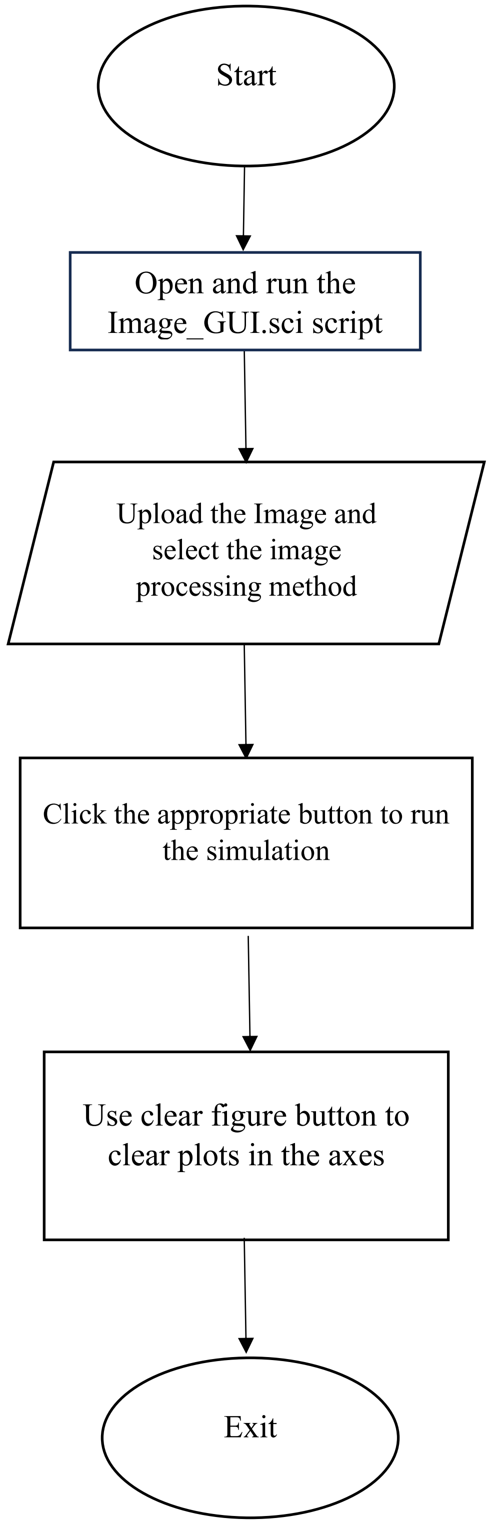
1. Initialization: Randomly select *k* initial centroids (cluster centres).
2. Assignment: Each data point is assigned to the nearest centroid, forming clusters.
3. Update: The centroids are recalculated as the mean (average) of all points within their respective clusters.
4. Iteration: Steps 2 and 3 repeat until the centroids stabilize (no significant change occurs).

The objective of K-Means is to group data points such that those within the same cluster are as similar as possible, while different clusters remain distinct. This method is widely used in image segmentation, customer segmentation, anomaly detection, and more due to its simplicity and efficiency.

3.2 Image segmentation based on Thresholding (Otsu's method)

Image segmentation using thresholding is a simple yet effective technique that separates objects from the background based on pixel intensity values. The method works by selecting a threshold value (T), where all pixels with intensities above T are classified as the foreground (white), while those below T become the background (black). This approach is particularly useful for high-contrast images with clear intensity differences between objects and their surroundings. There are several variations of thresholding, including global thresholding (using a single fixed threshold for the entire image), adaptive thresholding (where the threshold varies across regions to handle uneven lighting), and multi-level thresholding (using multiple thresholds to segment more complex images). While thresholding is computationally efficient and easy to implement, it has limitations, such as sensitivity to noise and lighting variations, which may require preprocessing steps like smoothing or contrast enhancement. Despite these challenges, thresholding remains widely used in applications like document processing, medical imaging, and object detection due to its simplicity and effectiveness in well-defined scenarios.

4 Flowchart



Start
Open and run the Image_GUI.sci script
Upload the Image and select the image processing method
Click the appropriate button to run the simulation
Use clear figure button to clear plots in the axes
Exit

5 Software/Hardware used

Operating System: Windows 11

Toolbox: Image Processing and Computer Vision (IPCV)

Hardware: Personal Computer with 12th Gen Intel Core Processor, 16GB RAM

Software: Scilab Version: 2025.0.0

6 Procedure of Execution

To generate the plots, follow these steps

Step 1: Launch Scilab on your computer and type editor in the Scilab console.

Step 2: Open the Image_GUI.sci file

Step 3: Run the file by selecting the triangular "Start" icon in the toolbar.

Step 4: Upload the image file from its current location.

Step 5: Select the image processing method and click the Run button to generate the appropriate plots in the right pane of the GUI interface. Each plot will display the simulated behaviour according to the selected configuration. To test different scenarios, modify the channel and system parameters in the GUI before running.

Step 6: Use the clear figure button to clear plots in the axes

Step 7: The Exit button closes the GUI window.

7 Results and Discussion

Figure 1 illustrates the comparative performance of the image segmentation algorithm using K-means clustering and thresholding algorithms on a sample butterfly image. The original and grayscale images serve as references, showing the intricate details and intensity variations present in the scene.

The first observation from the figure is that the segmented image using K-means demonstrates the algorithm's ability to partition the image into multiple regions based on colour similarity, though with some noise and scattered regions around the initial cluster centres. On the other hand, the thresholding result effectively highlights the prominent foreground object (the butterfly) against the background, providing a sharper and more distinct segmentation but at the cost of losing finer details and ignoring subtle intensity variations.

The second observation is that, with **k=2**, the K-means algorithm effectively divided the image into two clusters, broadly corresponding to the butterfly and the background. This choice of k simplifies the segmentation into a binary classification, similar to thresholding, but it still considers pixel intensity and colour similarity across the entire image, making it more adaptive to subtle variations than simple thresholding.

Overall, K-means is better suited for multi-region segmentation where colour variation is significant, whereas thresholding is more effective for binary foreground-background separation in high-contrast images.

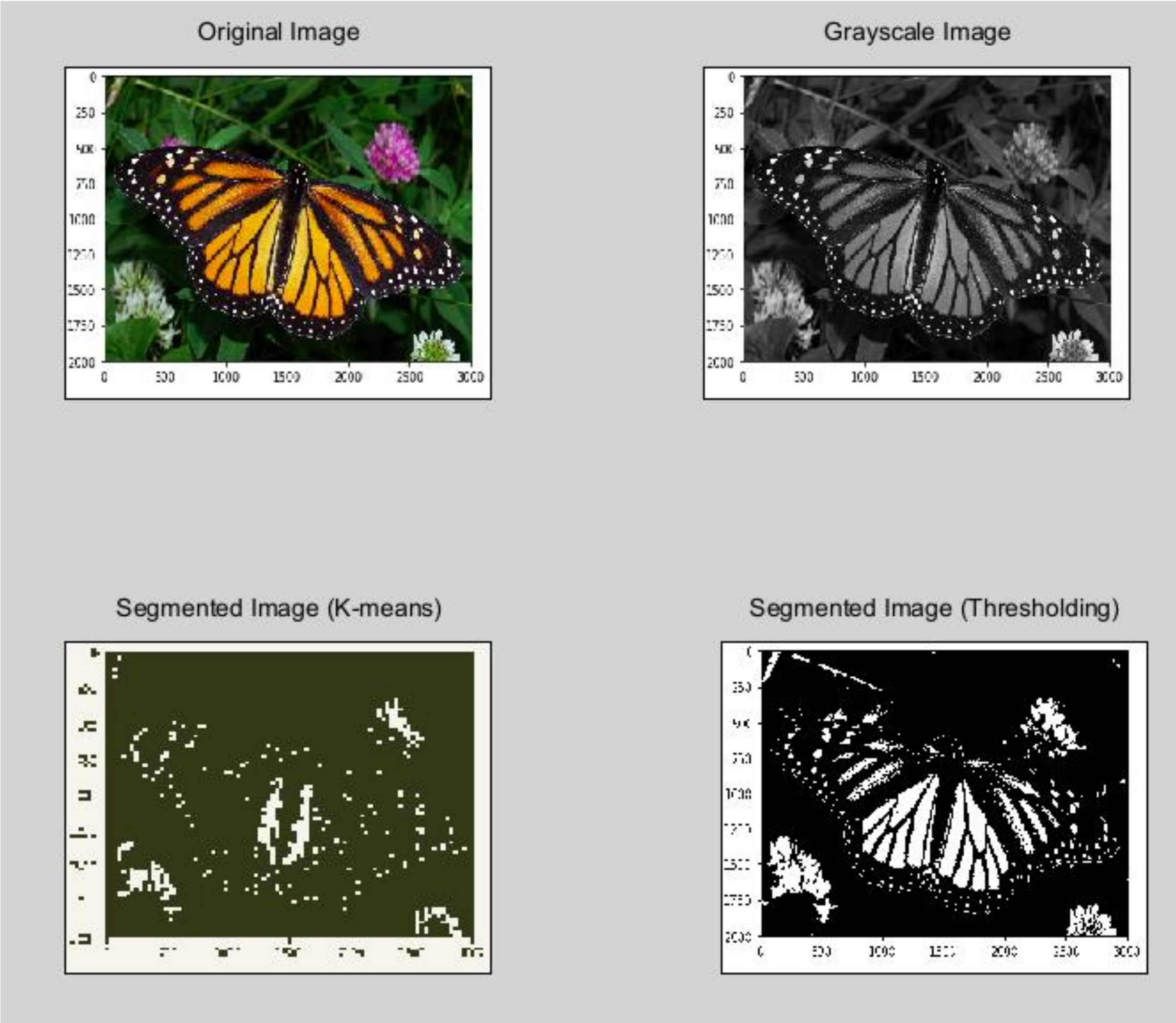


Figure 1: Comparative performance of the image segmentation algorithm using K-means (K=2) clustering and thresholding algorithms

Figure 2 demonstrates that the image segmentation over the butterfly image using K-means clustering with k=5 and a thresholding algorithm for comparison. When k=5, the K-means algorithm partitions the image into five distinct clusters based on colour similarity. The segmented image shows a more nuanced representation of the scene compared to k=2. Multiple shades of green, brown, and purple in the background, as well as different orange and black regions of the butterfly, are differentiated. This higher number of clusters allows finer details to emerge, preserving subtle texture and colour variations both in the object of interest and in the background.

In contrast, the thresholding output remains binary, with a clear separation of the butterfly from the background, but at the cost of finer details. From this observation, it can be inferred that choosing a higher k, such as 5, enables the capture of finer details and colour variations, which is beneficial for applications where texture and subtle differences matter (e.g., pattern analysis, medical imaging). On the other hand, it can also lead to over-segmentation and may require additional post-processing (e.g., region merging or filtering) to consolidate meaningful regions.

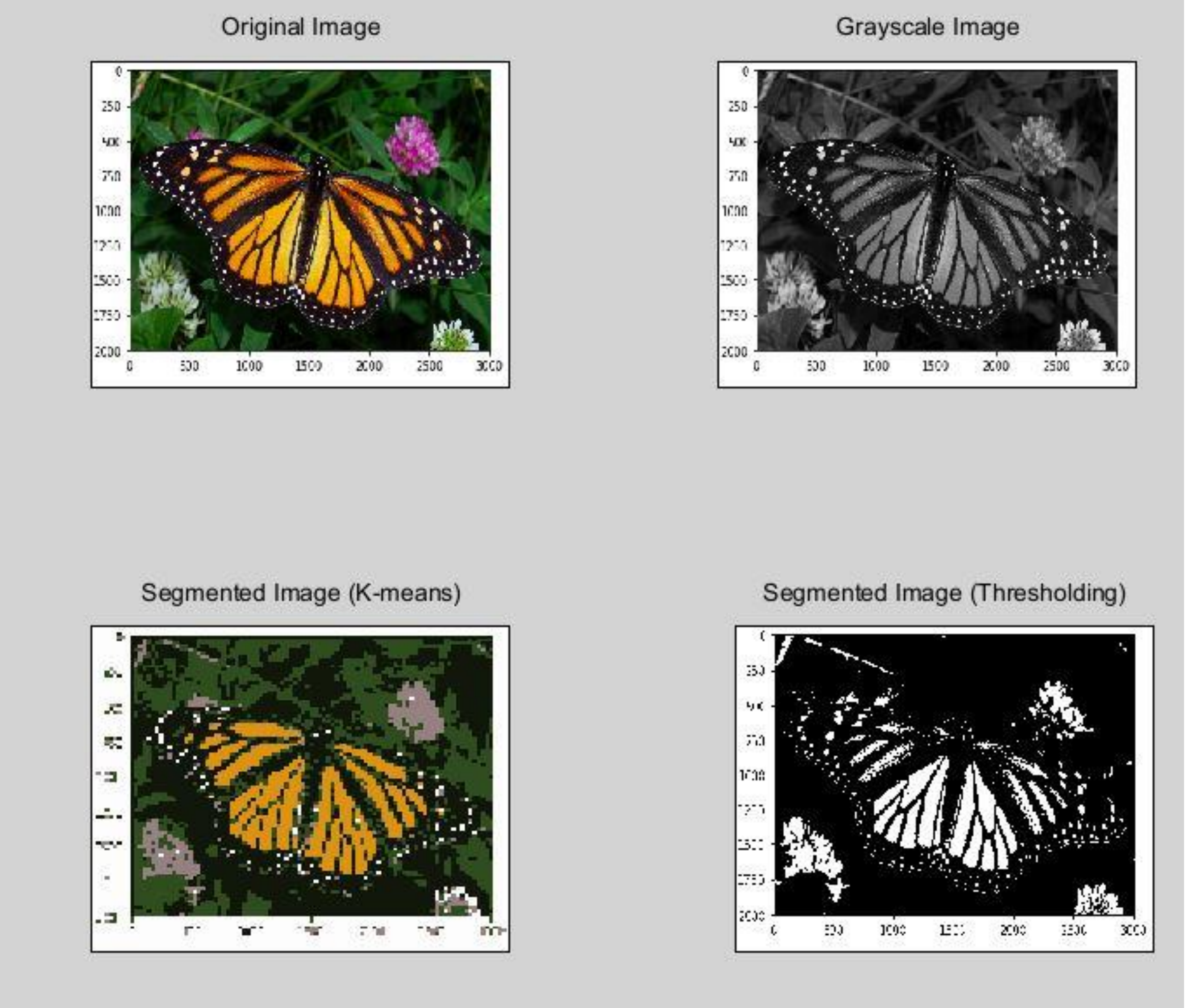


Figure 2: Comparative performance of the image segmentation algorithm using K-means (K=5) clustering and thresholding algorithms.

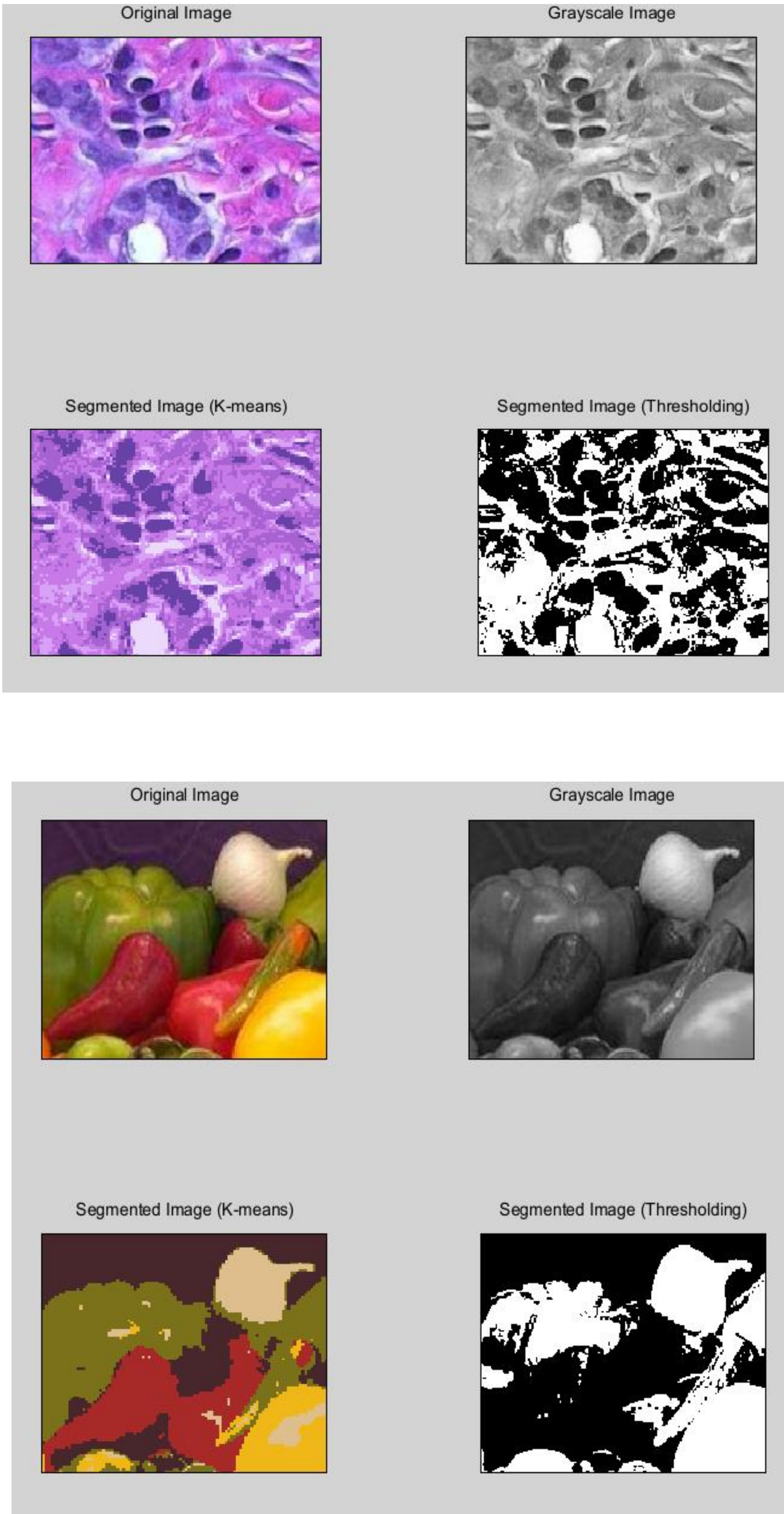


Figure 3: Comparative performance of the image segmentation algorithm using K-means clustering and thresholding algorithms with different images (Hestain and onion).

8 Reference

[1] Panwar, Preeti, Girdhar Gopal, and Rakesh Kumar. "*Image Segmentation using K-means clustering and Thresholding.*" Image 3, no. 05 (2016): 1787-1793.