Bananas Ripeness Level Using Brown Spots Detection

Sasha Annabel
Department of Computer Science
Yogyakarta, Indonesia

Maulana Arya Alambana
Department of Computer Science
Yogyakarta, Indonesia

Muhammad Rafi Cahya Ramadhana
Department of Computer Science
Yogyakarta, Indonesia

Raisa Aquila Zahra Kholiq
Department of Computer Science
Yogyakarta, Indonesia

***Abstract*—This paper presents a novel approach in the detection of ripeness level of bananas by detecting its brown spots through basic digital image processing techniques. The goal of this research is to develop an efficient and accurate system capable of quantifying the percentage of brown spots on banana surfaces, providing valuable insights into the ripeness of the fruit. The proposed methodology has a main process of image segmentation of banana regions using K-Means clustering techniques. Subsequently, the segmented banana areas undergo image enhancement and noise reduction to improve the accuracy of brown spot detection. The final stage involves quantifying the percentage of brown spots relative to the total banana area, providing a quantitative measure of ripeness. Experimental results demonstrate the effectiveness of the proposed method across diverse banana images, as well as the drawbacks of using the proposed methods. Nevertheless, the resulting program holds promise for applications in quality control within the agriculture and food industries, contributing to improved decision-making processes for ripeness assessment and quality assurance.**

Keywords—image processing, k-means clustering, segmentation

I. Introduction To Digital Image Processing

Image processing, an interdisciplinary domain at the intersection of computer science and visual perception, encompasses techniques that manipulate digital images. Its purposes vary for different types of images, with purposes of detecting objects or improving quality for further analysis.

Sections of image processing may include enhancement, segmentation, feature extraction, morphological processing, and others. From adjusting contrast and brightness to recognizing an object against its background, the methods are created in such diversity to accommodate for images of low to high pixel variance, with uneven or even spread of high pixel intensity concentration, with low to high visibility, or any distinct conditions. There are techniques to handle images with disruptive noise as well, by Median-Filtering and Convolution with Gaussian Blur for instance, as well as Thresholding for object and region segmentation. Whether processing the image in grayscale, or colored images in RGB or HSV, it is important to recognize the power these techniques carry to extract meaningful information from images, making it incredibly impactful and beneficial to this modern society that heavily relies on digital representation.

II. Problem Definition

The problem addressed in this paper is the development of an efficient and accurate system for the detection and quantification of brown spots on banana surfaces through basic digital image processing techniques. The challenges lie in achieving a reliable and automated method capable of precisely identifying and measuring the percentage of brown spots, a key indicator of banana ripeness. This paper specifically tackles issues related to image segmentation of banana regions, employing K-Means clustering techniques for this purpose. Additionally, challenges involve optimizing the accuracy of brown spot detection through image enhancement and noise reduction processes.

The underlying problem is to devise a method that not only effectively discerns brown spots but also quantifies their distribution relative to the total banana area, thereby providing a quantitative measure of ripeness. This paper addresses these challenges by presenting a novel approach, with experimental results showcasing its effectiveness across diverse banana images, while acknowledging and discussing potential drawbacks. The ultimate aim is to contribute to quality control in the agriculture and food industries, facilitating informed decision-making processes for ripeness assessment and quality assurance.

III. Problems During Solution Planning

There were deliberations on which image processing methods and platforms would be most suitable for the goal of detecting banana ripeness.

A. Method Considerations

The main priority is to figure out the main segmentation method, as it should have the ability to correctly identify which areas are considered as the 'brown spots', which area is just part of the 'yellow' banana area, and which is not part of the banana object itself (thus deemed as the background that is irrelevant).

Pixel-based segmentation methods are ruled out due to the color differences that can be tricky to detect. In our very first trial to test for segmentation using Thresholding, it is not rare for the program function to detect the brown spots in banana's skin as part of the background. Furthermore, the emphasis on color processing makes this method inefficient as it is more commonly used to segment binary (grayscale) images.

Feature-based segmentation methods are also not used due to the need for it to have high contrast between the feature and the Object Of Interest (OOI), which in this case may not be fully applicable. Though referred to with the term 'brown spots', they do not always have a dark brown color, and most often they have just a slight contrast with the banana area. Though faint, the difference is noticeable by human eyes, and thus the segmentation method should be able to handle such problems.

The final decision is to use the K-Means clustering method, which is hoped to handle color differences better by inputting the number of clusters that the program function should be able to detect. The K-Means algorithm is more independent against color ranges, meaning that the brown spots can be in any intensity as long as they are distinguishable against the normal yellow banana area, as well as the background.

B. Platform Considerations

The main platform we decided to use for this project is Google Colab. The main reason why Google Colab is used as the main platform is because of its numerous advantages. First of all, it offers free access to GPU resources, enhancing the speed and efficiency of computationally intensive tasks such as image segmentation and enhancement. This eliminates the need for costly local hardware and enables powerful computing capabilities without any significant financial investment. Additionally, Google Colab's cloud-based platform facilitates convenient collaboration by allowing multiple users to work simultaneously in real-time, eliminating the barriers associated with geographical constraints and ensures seamless team collaboration.

Another compelling reason to utilize Google Colab is its pre-installed libraries and dependencies, including popular tools like CV2, Numpy, SKlearn, and other libraries and dependencies. This eliminates the hassle of manual installations and configurations, allowing for direct project work. Moreover, Google Colab integrates seamlessly with Google Drive, simplifying data storage and retrieval. The platform's compatibility with Jupyter Notebooks, which is a widely adopted interface in data science, provides the users with a familiar and interactive environment for coding and debugging. With its scalability, accessibility, and easy integration with other Google Services, Google Colab emerges as a versatile and efficient choice for various digital image processing projects.

IV. Image Processing Methods Used

A. Image Acquisition

The Images used in the project required a few specifications to be able to detect the brown spots on the banana successfully. The main required specific characteristics to enable a successful detection of brown spot on bananas is by ensuring the distinctiveness of the banana from its background. To achieve this, we need to pay careful attention to lighting conditions, background uniformity, and overall image clarity.

B. Image Segmentation

As mentioned previously, the segmentation used for detecting brown spots in images of bananas will be K-Means clustering, whose unsupervised algorithm can segment the Areas Of Interest (AOI) from the background. The method will partition given image data into K-clusters or parts based on the *k* value written. A *k*=3 value would mean clustering the data into 3 parts, *k*=5 would mean clustering the data into 5 parts, and so on. The algorithm is highly effective in categorizing unlabeled data (i.e. data without defined categories or groups) such as in the case of this project where the yellow and brown colors to be detected in different banana images may not always have the same intensity (therefore cannot use a constant threshold).

C. Image Processing and Calculations

The clusters that are expected to be found will be the *background cluster* consisting of pixels that compose the background area (anything other than the banana object), the *brown spots cluster* that represents pixels of the brown spots in the banana skin, and the *yellow cluster* that represents the remaining area of banana skin. After the three above mentioned clusters are extracted, the main focus will be to do mathematical calculations on the *brown cluster* and *yellow cluster*. In precise, the aim is to find the percentage of brown spots area against the entire banana area. As such, the formula will be :

$$\frac{brown\_cluster\_area}{(brown\_cluster\_area + yellow\_cluster\_area)} \times 100$$

Formula (1)

Afterwards, our initial idea is to compare the brown spots percentage with a threshold of 40% :

- If percentage < 40% → ripe
- If percentage >= 40% → overripe

V. Solution Implementation

The code implementation is written in the main function called ***analyze_banana()*** that handles all segmentation methods and calculations within it.

A. First Run : Individual Banana Images with Brown Spots

The very first run of the ***analyze_banana()*** function proves that the proposed code works extremely well with images of individual bananas with brown spots.

1. Code Version 1

The main steps of the algorithm include :

- Apply K-means clustering to get three clusters
- Assign the cluster with least pixel as *brown_spots_cluster*, the cluster with second-least pixels as *yellow_cluster*, and the cluster with the most amount of pixels as *background_cluster*.
- Calculate the area of *brown_cluster* according to Formula (1).

```
def analyze_banana(image_path, color_difference_threshold=0.3):
    # Read and reshape the image
    pic = plt.imread(image_path) / 255  # dividing by 255 to bring the pixel values between 0
and 1
    pic_n = pic.reshape(pic.shape[0] * pic.shape[1], pic.shape[2])

    # Apply K-means clustering
    kmeans = KMeans(n_clusters=3, random_state=0).fit(pic_n)
    labels = kmeans.labels_

    # Reshape the labels to match the original image shape
    labels_reshaped = labels.reshape(pic.shape[0], pic.shape[1])

    # Display unique cluster labels and their counts
    unique_labels, counts = np.unique(labels, return_counts=True)
    print("Cluster Labels and Counts:")
    for label, count in zip(unique_labels, counts):
        print(f"Cluster {label}: {count} pixels")

    # Find the labels of the clusters with the least and second-least pixels
    least_pixels_cluster = unique_labels[np.argmin(counts)]
    second_least_pixels_cluster = unique_labels[np.argsort(counts)[1]]
    print(f"Cluster with Least Pixels: {least_pixels_cluster}")
    print(f"Cluster with Second-Least Pixels: {second_least_pixels_cluster}")

    # Find the indices of the least and second-least pixels in the labels
    least_pixels_indices = np.where(labels == least_pixels_cluster)
    second_least_pixels_indices = np.where(labels == second_least_pixels_cluster)

    # Extract the colors of the least and second-least pixels
    least_pixels_colors = pic_n[least_pixels_indices]
    second_least_pixels_colors = pic_n[second_least_pixels_indices]

    # Calculate the color difference between the two clusters using RGB values
    color_difference = calculate_rgb_distance(np.mean(least_pixels_colors, axis=0),
np.mean(second_least_pixels_colors, axis=0))
    print(f"Color Difference Between Least and Second-Least Pixels: {color_difference}")


    # Determine which is the brown_spots_cluster and which is the yellow_cluster
    brown_spots_cluster = least_pixels_cluster
    yellow_cluster = second_least_pixels_cluster

    # Calculate area of the yellow area and brown spots area
    yellow_area = np.sum(labels_reshaped == yellow_cluster)
    brown_spots_area = np.sum(labels_reshaped == brown_spots_cluster)

    # Calculate total area of the image -- not really needed, just in case
    total_area = pic.shape[0] * pic.shape[1]

    # Calculate percentage of brown spots over the BANANA (yellow area + brown spots)
area
    percentage_brown_spots_over_yellow = (brown_spots_area / (yellow_area +
brown_spots_area)) * 100 if yellow_area != 0 else 0.0

    if percentage_brown_spots_over_yellow <= 40:
      ripeness = 'Ripe'
    else:
      ripeness = 'Overripe'

    # Visualization
    pic2show = kmeans.cluster_centers_[kmeans.labels_]
    cluster_pic = pic2show.reshape(pic.shape[0], pic.shape[1], pic.shape[2])
    plt.imshow(np.hstack((pic, cluster_pic)))
    plt.title('Original Image | Segmented Image')
    plt.axis('off')
    plt.show()

    return yellow_area, brown_spots_area, percentage_brown_spots_over_yellow, ripeness
```

Code Snippet 1.a

2. Results

a) Successful Testcase 1 : High Brown Spots Area

The proposed ***analyze_banana()*** function has been proven to handle well for recognizing and calculating the brown spots area when the percentage area is high, about 36%.

Figure 1.a

Table for Figure 1.a	
Cluster 0	179363 px
Cluster 1	27267 px
Cluster 2	47350 px
Cluster for Brown Spots (Cluster with least pixels)	Cluster 1
Cluster for Yellow Area (Cluster with second-least pixels)	Cluster 2
Color difference between clusters with least and second-least pixels	0.52123240130
Percentage of Brown Spots Area	**36.542610%**
Classification	**Ripe**

b) Successful Testcase 2 : Low Brown Spots Area

The proposed ***analyze_banana()*** function has also been proven here to handle well for recognizing and calculating the brown spots area when the percentage area is low, about 11 %.

Figure 1.b

Table for Figure 1.b	
Cluster 0	72184 px
Cluster 1	232918 px
Cluster 2	9842 px
Cluster for Brown Spots (Cluster with least pixels)	Cluster 2
Cluster for Yellow Area (Cluster with second-least pixels)	Cluster 0
Color difference between clusters with least and second-least pixels	0.43868258961
Percentage of Brown Spots Area	**11.9986345%**
Classification	**Ripe**

c) *Unsuccessful Testcase 3 : Shadow In Image*

The ***analyze_banana()*** function with K-Means algorithm has a much lower accuracy when the lighting of the banana in the image is not even. When shadow is present around the banana area, it has a very high chance to be included in the cluster of brown spots, despite it not being a brown spot itself. As such, it is a newfound limitation of the program; to acquire images of bananas that must have even lighting conditions.

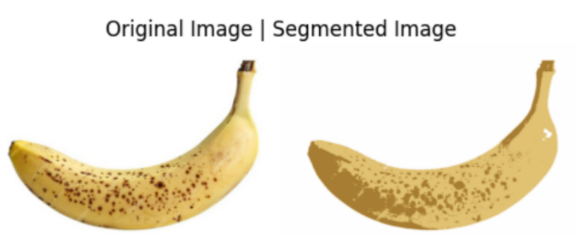

Figure 1.c

Table for Figure 1.c	
Cluster 0	247903 px
Cluster 1	64709 px
Cluster 2	31296 px
Cluster for Brown Spots (Cluster with least pixels)	Cluster 2
Cluster for Yellow Area (Cluster with second-least pixels)	Cluster 1
Color difference between clusters with least and second-least pixels	0.4329112186
Percentage of Brown Spots Area	**32.598302%**
Classification	**Ripe**

B. *Second Run : Individual Banana Images with No Brown Spots*

The **analyze_banana()** function seems to not work as intended in banana images with no brown spots area at all. The problem found is when the proposed algorithm attempts to categorize different shades of yellow as different clusters, due to the absence of brown spots in the entire banana skin area. In this particular case, the K-Means algorithm still attempts to find 3 clusters, though no brown areas are visible. Instead of getting three clusters consisting of background, yellow area, and brown area, the K-Means algorithm takes instead the area of a darker/lighter shade of yellow in replacement of the absent brown spots area.

As such, it is necessary to have a **code-check** where the two clusters (remember; *background_cluster* is ignored) are examined for their pixel color differences. The idea is, if the pixel color difference is lower than a certain threshold, then it can be assumed that the two clusters are actually very similar in color in their original image.

1. Code Version 2

```
... (previous code sections remain unchanged)
 # The two lines below are ADDED AFTER Failed Case 2
  if color_difference < color_difference_threshold :
    plt.imshow(pic)
    plt.title('Original Image')
    plt.axis('off')
    plt.show()
    print(f"Percentage of brown spots may be less than 3%. Detection limited.")
    return least_pixels_indices + second_least_pixels_indices, 0, 3, 'Newly Ripe'
... (next  code sections remain unchanged)
```

Code Snippet 1.b

The additional section of the code handles situations where the two non-background clusters are, in actuality, both supposed to represent the yellow area. If the pixel color difference is higher than the threshold of 0.3, then it is considered safe to assume that the two clusters are of very distinct colors, or in this case, are of yellow area and brown spots area. Otherwise, if the pixel color difference is less than 0.3, then the two clusters are both considered yellow, just in different shades.

2. **Results**

Figure 2.a shows the way the algorithm attempts to segment the banana area before Code Snippet 1.b is added into the ***analyze_banana()*** function. As seen from the Table for Figure 2.a, Cluster 1 is referred to as Cluster for Brown Spots, even though there are no brown spots area present. In this particular case, the two non-background clusters are simply two different shades of yellow.

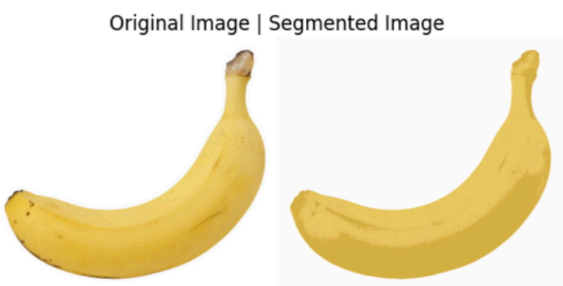

Figure 2.a

Table for Figure 2.a	
Cluster 0	3349011 px
Cluster 1	636621 px
Cluster 2	773920 px
Cluster for Brown Spots (Cluster with least pixels)	Cluster 1
Cluster for Yellow Area (Cluster with second-least pixels)	Cluster 2
Color difference between clusters with least and second-least pixels	0.1822822987
Percentage of Brown Spots Area	**45.133108%**

Though the code still finds the clusters the same as seen in Table for Figure 2.a after Code Snippet 1.b is added into the analyze_banana() function, it checks for the pixel color difference before returning the percentage of brown spots area. Since the pixel color difference is found to be 0.18, which is lower than the determined threshold of 0.3, then the two clusters are assumed to be of the same color (both represent the yellow area). The output for the brown spots area then, is no longer printed as "45.133108%", but instead as :

Percentage of brown spots is less than 3%. Detection limited.

The banana is also categorized as 'Newly Ripe', a new category that was initially not planned. Since it will be difficult to determine the area of brown spots when the area is too little for K-Means to even detect their presence as a cluster on their own, it is assumed that the area of the brown spots is either nonexistent or less than 3%. This is another newfound limitation of the proposed algorithm.

C. Third Run : Group Banana Images with Brown Spots

This third run of the ***analyze_banana()*** function proves that the proposed algorithm works well with images of bananas where there are more than one banana in one same image.

1. **Results**

Figure 3.a

Table for Figure 3.a	
Cluster 0	66654 px
Cluster 1	25546 px
Cluster 2	22031 px
Cluster for Brown Spots (Cluster with least pixels)	Cluster 2
Cluster for Yellow Area	Cluster 1

(Cluster with second-least pixels)	
Color difference between clusters with least and second-least pixels	0.5137278522
Percentage of Brown Spots Area	**46.305988%**
Classification	**Overripe**

It is important to note, however, that though the algorithm works, the results are in lower accuracy than for images with only one single banana. This is due to possible shadows between bananas that cover each other, and these shadows being considered as brown spots.

D. *Initial Results Discussion*

Our proposed algorithm, which particularly relies on the K-Means clustering for segmentation, proves to be effective, yet it must be kept in mind that there are special cases, or criteria, in analyzing the banana images.

- *The background of the images must be in a single color.*
  The authors of this paper assumed that the background, which is in one color, would always be categorized in the same cluster (thus, as explained before, the *k* value is constant 3). This assumption is not realistic and is only made for simplicity, as background images tend to be in multiple colors and lighting conditions are very high. In such circumstances, the clusters may have to be more than the initial 3.
- *The banana images must not have shadows.*
  Having shadows, particularly around the banana area, results in the algorithm categorizing the shadow area as part of the brown spots area. This condition will lead to skewed data for the brown spots area, making its percentage over the banana area higher than in actuality. This includes cases for images that have more than one individual banana, where shadows might present more likely depending on the lighting.
- *Limited detection for <3% brown spots area.*
  The K-Means algorithm has drawbacks, where if the brown spots are either too little or unable to be 'discovered', then it searches for other colors to cluster. In other words, the algorithm cannot be informed to specifically dedicate a cluster for the brown spots area, since it is in its nature to cluster unlabeled groups with distinct features automatically and flexibly. It is then the decision to make a special output for cases where the brown spots area covers less than 3% of the banana area.

E. *Overall Quantitative Outcome Analysis*

To test more properly the effectiveness of our proposed algorithm, we experimented on 12 more cases as seen in the table below. It is important to keep in mind, the authors are unable to provide the expected percentage of brown spots for each image to compare with the algorithm's findings. As such, the comparison focuses more on simply the final classification output of the image (whether Newly Ripe, Ripe, or Overripe).

Table for Quantitative Analysis

No	Image	Expected Classification	Algorithm Output Classification
1		Ripe	Ripe
2		Ripe	Ripe
3		Newly Ripe	Newly Ripe
4		Newly Ripe	Ripe
5		Overripe	Ripe
6		Ripe	Ripe
7		Ripe	Ripe

8		Ripe	Ripe
9		Ripe	Ripe
10		Ripe	Ripe
11		Overripe	Overripe
12		Overripe	Overripe

According to the results of our 12 trials, 2 of the 12 banana images had the results incorrect to the expected outcome. As such, the current quantitative success rate of our proposed ***analyze_banana()*** algorithm is about 83.3%. The percentage results, which can be directly observed below the *Additional Testcases* section at the Google Colab platform, are more complicated to analyze. Certain images, such as Image No. 7 and No. 8, have percentage results that the authors believed to be skewed due to the little presence of shadows and poor lighting conditions, though they nevertheless led to the desired expected outcome.

VI. Conclusion

Although the ***analyze_banana()*** algorithm is working as intended, there are criteria for the initial input images as well as limitations in their results detection. The found problems can perhaps be solved by applying other image processing methods on top of the proposed solution, whether adding pre-processing or other segmentation methods.

Acknowledgment

The authors thank Mr. Wahyono for his teachings in Digital Image Processing and guidance, which has made this research possible.

References

[1] Shiv Ram Dubey, Pushkar Dixit, Nishant Singh, Jay Prakash Gupta, “Infected Fruit Part Detection using K-Means Clustering Segmentation Technique”, International Journal of Artificial Intelligence and Interactive Multimedia, Vol. 2, Nº 2.

[2] Mayuri Wankhade, U. W. Hore, “Banana Ripeness Classification Based On Image Processing With Machine Learning”, International Journal of Advanced Research in Science, Communication and Technology (IJARSCT) Volume 6, Issue 2, June 2021.

[3] AshalKamdar, “Image Segmentation using K Means Clustering”, https://www.geeksforgeeks.org/image-segmentation-using-k-means-clustering/

[4] I. A. Sabilla, C. S. Wahyuni, C. Fatichah and D. Herumurti, "Determining Banana Types and Ripeness from Image using Machine Learning Methods," 2019 International Conference of Artificial Intelligence and Information Technology (ICAIIT).

[5] P. K. Devi and Rathamani, "Image Segmentation K-Means Clustering Algorithm for Fruit Disease Detection Image Processing," 2020 4th International Conference on Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, India, 2020.

[6] Shiv Ram Dubey, Pushkar Dixit, Nishant Singh, Jay Prakash Gupta, “Infected Fruit Part Detection using K-Means Clustering Segmentation Technique”, International Journal of Artificial Intelligence and Interactive Multimedia, Vol. 2, Nº 2.