Banana Ripeness Detection System Using Image Segmentation and Rule-Based Color Analysis

Group 13

Mohammad Ahmed Abdullah Ismail
Huzaifa Muhammad Siddique
Ahmed M. A. Albreem
Mustafa Kamal Mustafa Mustafa

December 17, 2025

Abstract

This project focuses on automatically detecting the ripeness of a banana using image processing and rule-based color analysis. The goal is to replace manual fruit inspection, which is slow, inconsistent, and expensive, with a faster and more reliable automated system. The method begins by converting each banana image from the RGB color space to the HSV color space, which makes it easier to analyze color changes related to ripeness. The Saturation channel is smoothed using Gaussian smoothing and processed using Otsu thresholding to create a mask that isolates the banana from the background. Morphological operations are then applied to clean the mask and preserve only the banana region.

After segmentation, ripeness is determined using only banana pixels by measuring green, yellow, and dark-spot ratios from the Hue and Value channels. A simple rule-based classifier then labels the banana as unripe, ripe, or overripe based on these ratios. The system is tested on a collection of banana images, and all results are stored in a CSV file for evaluation. This approach can be applied in real-world scenarios such as automated sorting lines, retail inventory systems, and agricultural monitoring, helping to reduce food waste and improve efficiency.

1 Introduction

The problem that this project addresses is detecting the ripeness of a banana from an image by analyzing the color of the banana. The proposed system combines image segmentation, signal processing techniques, and rule-based color analysis to determine ripeness levels in an automated manner.

There are several motivations behind this banana ripeness detection project. The primary motivation is to automate quality control using an image-based system, which is especially beneficial at large scales. Fruit inspection in farms and packaging facilities is still largely manual, making it slow, inconsistent, and expensive when performed extensively. An automated inspection system is faster, more reliable, continuously operational, and significantly more cost-effective.

Another important motivation is reducing fruit waste, particularly for bananas. Bananas are among the most wasted fruits due to their rapid and inconsistent ripening behavior. An automatic ripeness detection system can help supermarkets monitor banana freshness and enable farmers to determine optimal harvesting and shipping times. This contributes directly to minimizing food waste.

The real-world applications of this project are broad and highly relevant to both agricultural and industrial domains. Some key application areas include:

1. Automated Sorting Lines in Banana Packaging Facilities
   Commercial banana farms sort fruit into different categories before export. The proposed system can automatically classify bananas by ripeness level and route them to the appropriate shipping containers, significantly reducing labor costs.

2. Smart Grocery Stores and Retail Inventory Management
   Retailers can use the system to assess banana ripeness on store shelves, apply discounts before the fruit becomes unsellable, and automatically reduce food wastage.

3. Precision Agriculture and Farm Monitoring
   Farmers can analyze ripeness trends across different harvest batches using ground-level imagery. This enables improved harvesting schedules, optimized shipping times, and better storage planning.

4. Robotics and Automated Picking Systems
   Banana-picking robots require real-time decisions regarding whether a banana should be harvested. The proposed ripeness detection system can assist robots in making these decisions and provides a foundational approach for automated fruit harvesting applications.

2 Methodology

The proposed system begins by processing an input image containing a banana. Each image is initially represented in the RGB color space; however, RGB is not well suited for analyzing color variations associated with fruit ripeness. Therefore, the image is converted to the HSV color space, which separates color information into Hue (color), Saturation (color intensity), and Value (brightness). This representation allows more effective analysis of ripeness-related color changes.

To reduce noise and small pixel-level variations, Gaussian smoothing is applied to the Saturation channel. This filtering step suppresses high-frequency noise while preserving the overall structure of the image, resulting in more stable and accurate segmentation.

Banana segmentation is then performed using thresholding techniques on the Saturation channel of the HSV image. The Saturation channel is chosen because banana pixels typically exhibit higher saturation values than a white or light-colored background. Thresholding converts the saturation image into a binary mask, where pixels above a threshold are classified as foreground (banana) and the remaining pixels are treated as background. However, this step alone may produce inaccurate results when the background contains strong colors.

To address this, an automatic threshold is computed using Otsu's algorithm. Pixels with saturation values above the computed threshold are classified as banana pixels, while the remaining pixels are considered background.

After thresholding, the resulting binary mask may contain small gaps within the banana region or noisy regions outside of it. Morphological operations are applied to remove noise and fill gaps. Finally, only the largest connected component is retained as the banana mask. This ensures that background pixels are excluded from further processing, reducing their influence on subsequent measurements.

Once the banana mask is obtained, ripeness analysis is performed exclusively on the pixels belonging to the banana. This guarantees that background colors do not affect the ripeness estimation.

Ripeness determination is carried out using a rule-based classification approach. Green and yellow pixel proportions are measured using the Hue channel, while dark pixel proportions (representing spots) are identified using low values in the Value channel. If the proportion of green pixels exceeds a predefined threshold, the banana is classified as unripe. If the proportion of dark pixels exceeds a predefined threshold, the banana is classified as overripe. If neither condition is satisfied, the banana is classified as ripe.

3 Data and Implementation

3.1 Dataset Description

The dataset consists of banana images collected from online sources. Each image contains a single banana and is stored in one of two directories: `data/dev/` and `data/eval/`. To maintain consistency and enable easy batch processing using scripts, all images were renamed using a uniform naming convention.

3.2 Libraries Used

The project is implemented using the following Python libraries:

- **scikit-image (skimage):** Used for image input/output (`skimage.io`), color space conversion (`skimage.color`), filtering and thresholding (`skimage.filters`), morphological operations (`skimage.morphology`), and connected component analysis (`skimage.measure`).
- **NumPy:** Used for numerical operations such as pixel counting, masking, and computing color ratios.
- **SciPy (scipy.ndimage):** Used for hole filling in binary masks using `binary_fill_holes`.

3.3 Code Structure and Implementation Details

3.3.1 Banana Segmentation Module — `src/segment.py`

This module contains the segmentation function `segment_banana_mask()`, which takes an RGB image as input and returns a binary mask of the banana region (`True` for banana pixels and `False` for background pixels). The segmentation approach is based on converting the image to the HSV color space and thresholding the Saturation channel, as banana regions typically have higher saturation than a white background.

A portion of the implementation, including HSV conversion, Gaussian smoothing, and Otsu thresholding, is shown below:

```
hsv = color.rgb2hsv(rgb_f)
s = hsv[:, :, 1]
s_smooth = filters.gaussian(s, sigma=sigma)

t = filters.threshold_otsu(s_smooth)
mask = s_smooth > t
```

Listing 1: HSV Conversion, Smoothing, and Otsu Thresholding

After thresholding, the binary mask is refined using morphological operations and hole filling to remove noise and close small gaps in the banana region. Since each image contains only one banana, connected component analysis is applied to retain only the largest connected region as the final banana mask.

3.3.2 Batch Segmentation Runner — `scripts/run_segmentation_batch.py`

This script performs batch segmentation on all images located in `data/dev/` and `data/eval/`. It iterates over each image file, generates the banana mask, and saves the resulting masks and overlay visualizations to the following directories:

- `outputs/masks/dev/`
- `outputs/masks/eval/`

- `outputs/preview/dev/`
- `outputs/preview/eval/`

3.3.3 Ripeness Analysis Module — `src/ripeness.py`

This module contains the feature extraction and rule-based classification logic. After segmentation, ripeness analysis is performed exclusively on banana pixels, ensuring that background colors do not influence the measurements.

The following code illustrates how HSV values are extracted only from banana pixels:

```
hsv = color.rgb2hsv(util.img_as_float(rgb))
H, S, V = hsv[:, :, 0], hsv[:, :, 1], hsv[:, :, 2]

H_b = H[banana_mask]
S_b = S[banana_mask]
V_b = V[banana_mask]
```

Listing 2: Extracting HSV Values from Banana Pixels

Using these values, several color ratios are computed, including the green ratio, yellow ratio, and dark spot ratio. An example of the main pixel-counting logic is shown below:

```
colored = S_b > sat_min
green = colored & (H_b >= green_lo) & (H_b <= green_hi)
dark = V_b < dark_v_thresh

green_ratio = np.sum(green) / H_b.size
dark_ratio = np.sum(dark) / H_b.size
```

Listing 3: Pixel Ratio Computation

Finally, a rule-based classifier assigns a ripeness label based on the computed ratios:

```
if green_ratio > 0.30:
    label = "unripe"
elif dark_ratio > 0.08:
    label = "overripe"
else:
    label = "ripe"
```

Listing 4: Rule-Based Ripeness Classification

3.3.4 Batch Ripeness Runner — `scripts/run_ripeness_batch.py`

This script executes the complete pipeline, including segmentation, feature extraction, and classification, for all images in the development and evaluation sets. The final ripeness predictions are saved to a CSV file:

- `outputs/predictions.csv`

This file is later used for evaluation and result analysis.

4 Results and Discussion

Since the dataset used in this project is unlabeled, it is not possible to compute a numerical accuracy metric, as there are no ground-truth labels for comparison. However, extensive manual inspection was performed by visually comparing the predicted ripeness labels with the actual appearance of the bananas. Based on this inspection, the system demonstrated excellent performance and was found to be highly consistent with human judgment.

Throughout the development process, several visualizations were generated to analyze and justify the choice of segmentation and color-based methods. In particular, histograms and color-space visualizations were examined to determine whether RGB or HSV representations were more suitable. Although alternative and more complex methods exist, they were considered outside the scope of this project.

4.1 RGB Color Space Analysis

Figure 1 shows an RGB image of a banana with a yellow background, which is not within the intended scope of the project but is included for comparison.

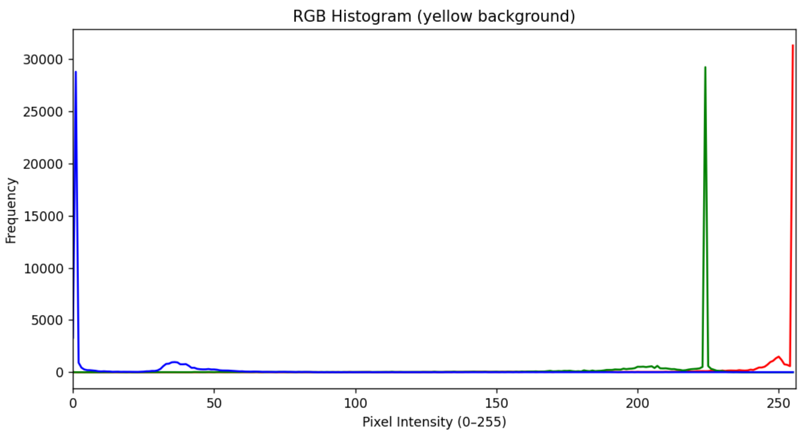

Figure 1: RGB image for a banana with yellow background

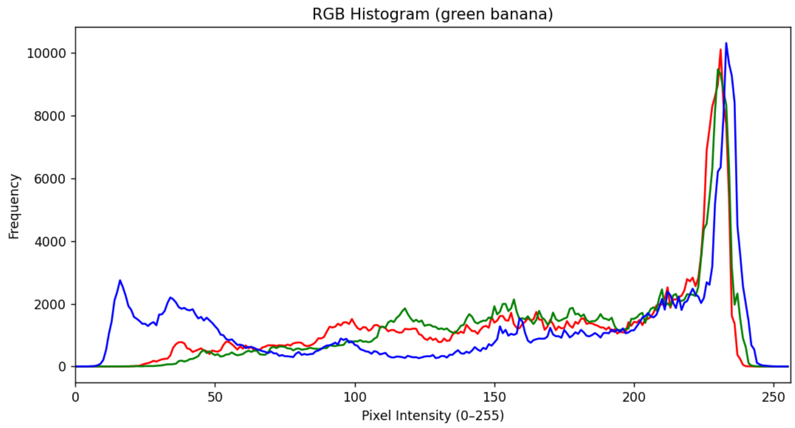

Figure 2: RGB image for a green banana with white background

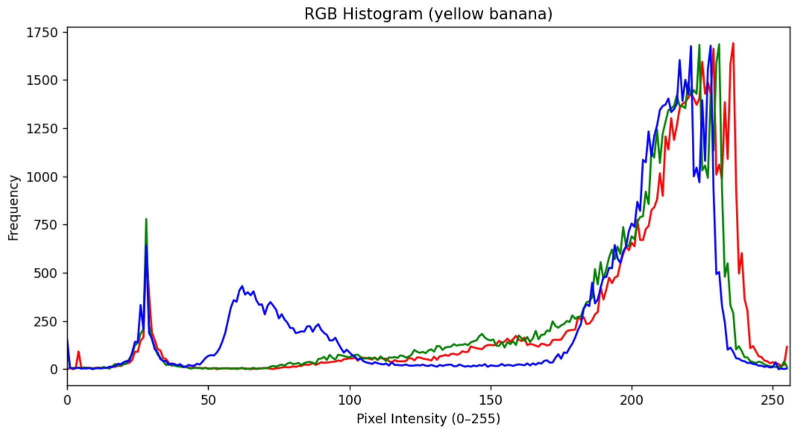

Figure 3: RGB image for a yellow banana with white background

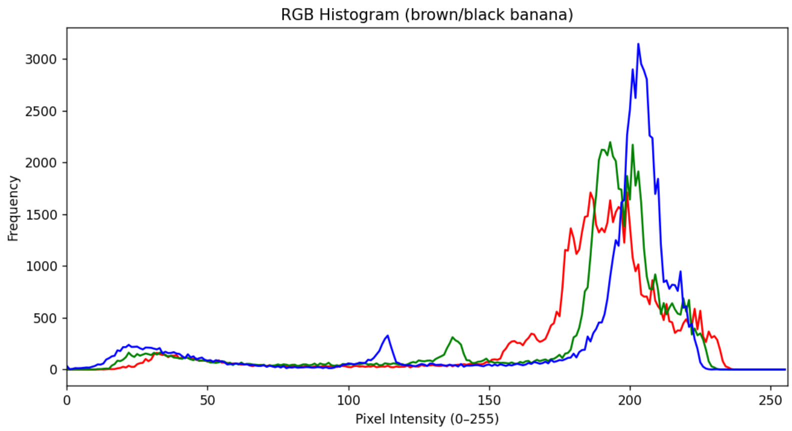

Figure 4: RGB image for a brown/black banana with white background

4.2 HSV Color Space Analysis

The same set of bananas is visualized in the HSV color space to highlight how hue, saturation, and value separate ripeness-related features more effectively.

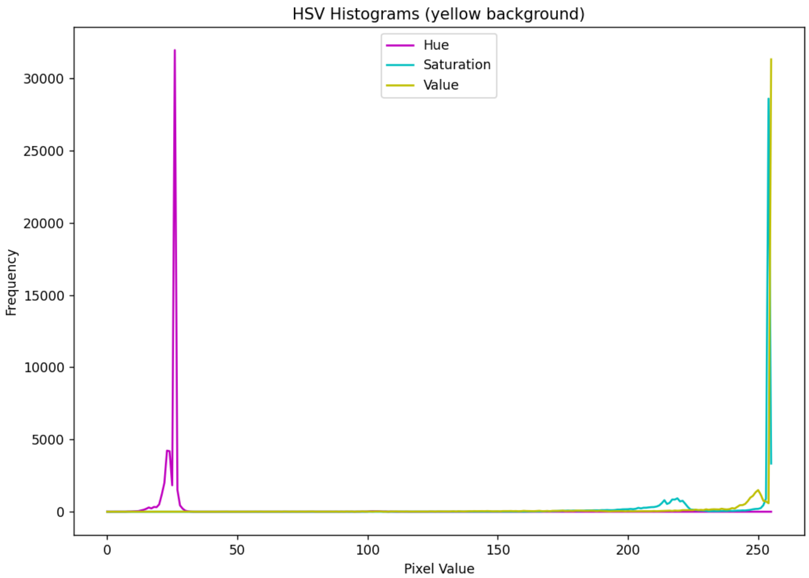

Figure 5: HSV image for a banana with yellow background

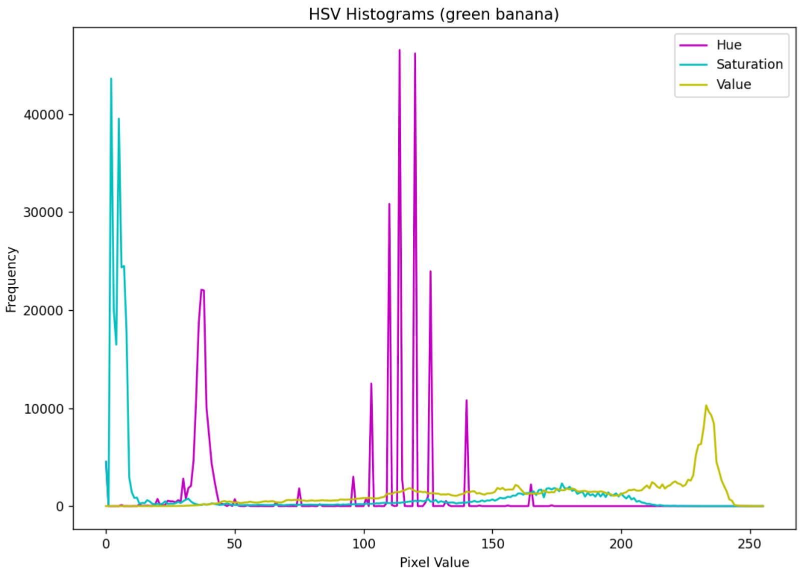

Figure 6: HSV image for a green banana with white background

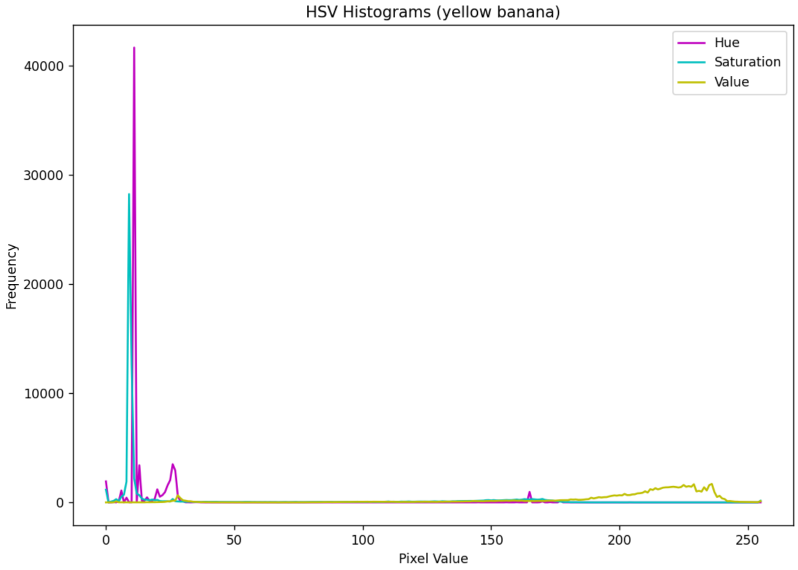

Figure 7: HSV image for a yellow banana with white background

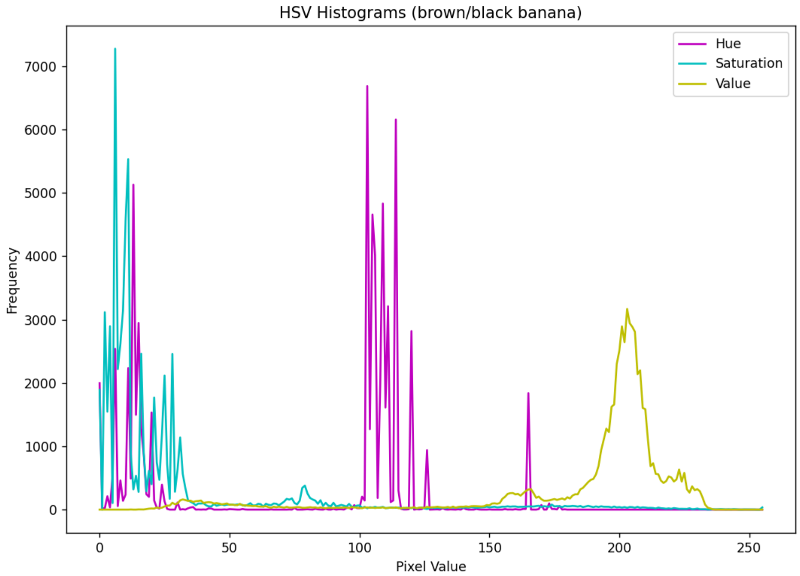

Figure 8: HSV image for a brown/black banana with white background

4.3 Feature Distribution and Correlation Analysis

To further analyze ripeness behavior, mean hue and mean value scores were computed for all bananas and visualized in scatter plots.

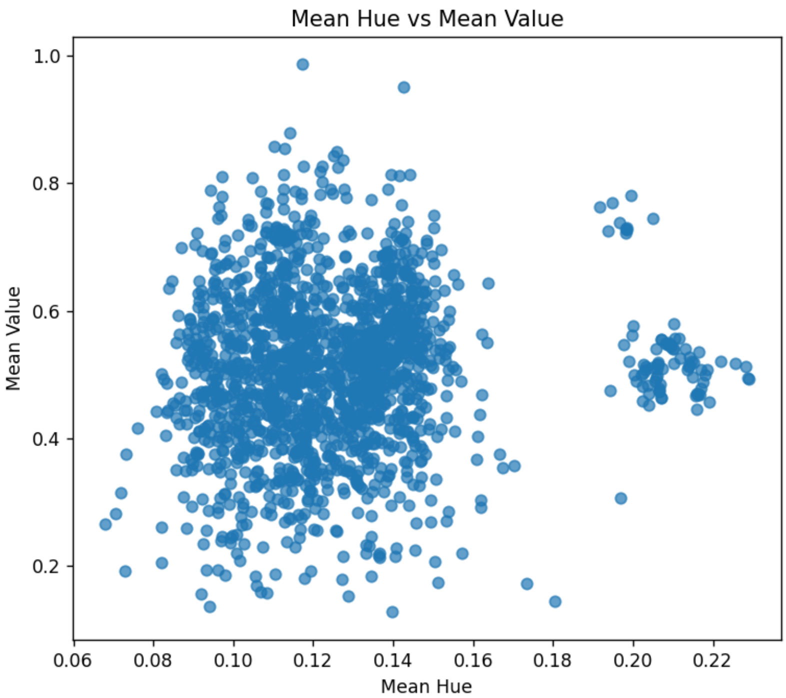

Figure 9: Mean Hue vs Mean Value

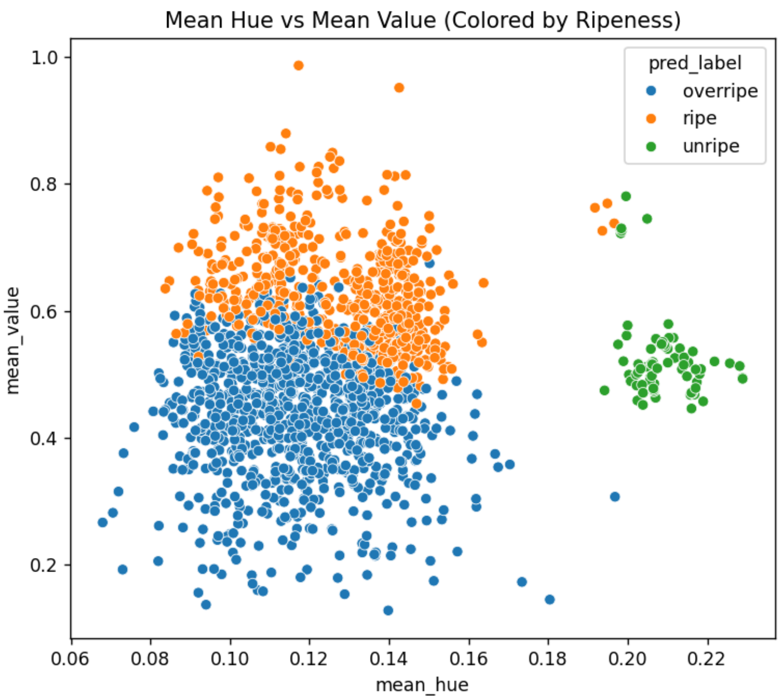

Figure 10: Mean Hue vs Mean Value (Colored by Ripeness)

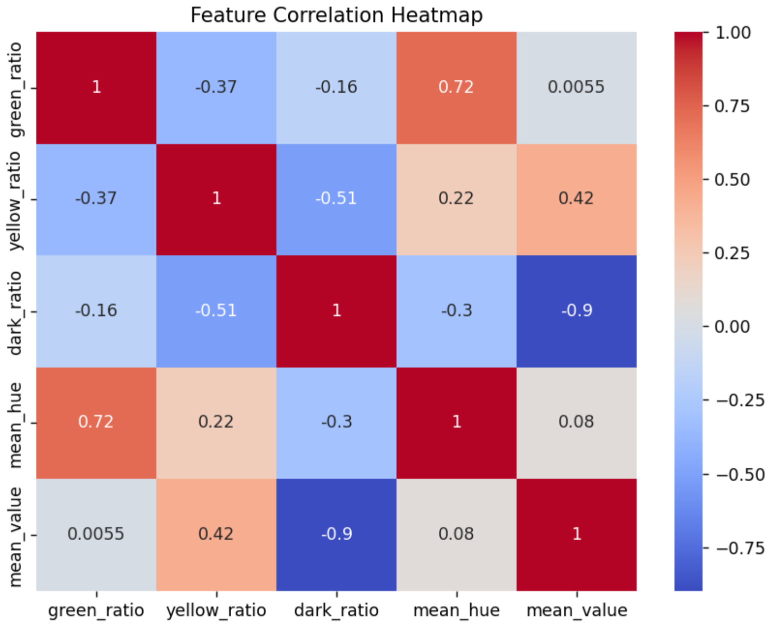

Figure 11: Feature Correlation HeatMap

4.4 Final Ripeness Distribution and Ratios

The final predicted ripeness labels are summarized in Figure 12, showing the distribution across unripe, ripe, and overripe categories.

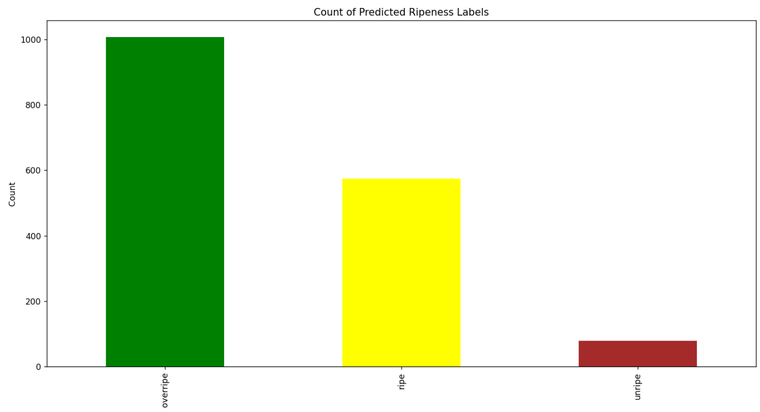

Figure 12: Count of Predicted Ripeness Labels

Figures 13, 14, and 15 show the ratios of green, yellow, and brown/black pixels across the dataset.

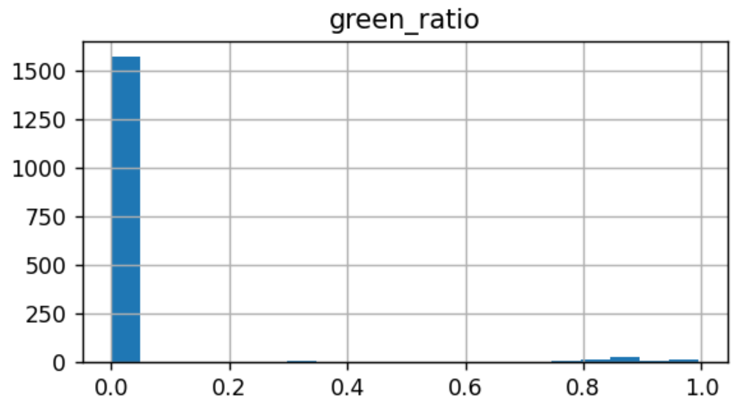

Figure 13: Ratio of Green Bananas Against Their Count

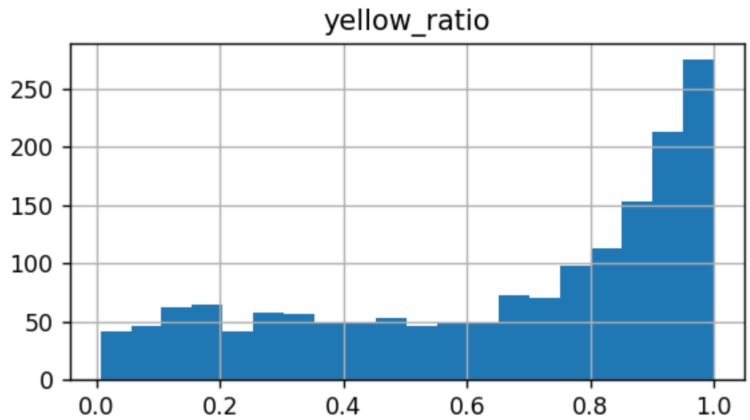

Figure 14: Ratio of Yellow Bananas Against Their Count

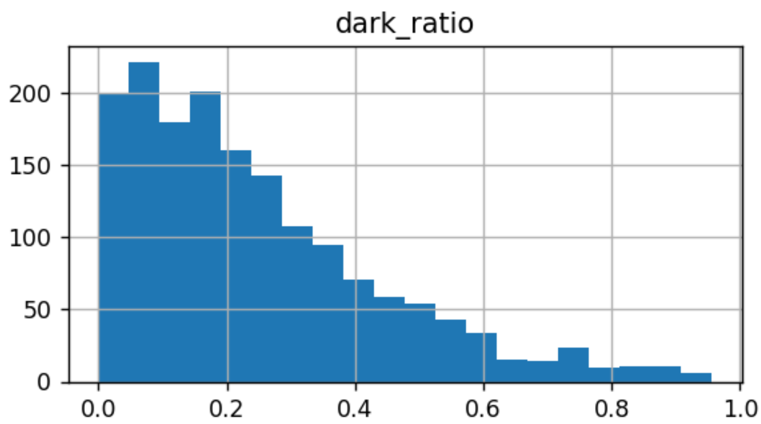

Figure 15: Ratio of Brown/Black Bananas Against Their Count

4.5 Qualitative Result Visualization

Here are three example screenshots of the final output produced by the proposed system. Each image displays the original banana, the segmented banana mask, and the banana with highlighted red regions used for ripeness analysis. One example is shown for each ripeness category: green, yellow, and brown/black bananas.

Figure 16: Result for green banana

Figure 17: Result for yellow banana

Figure 18: Result for brown/black banana

5 Conclusion

In this project, a banana ripeness detection system was developed using image segmentation and rule-based color analysis. The system isolates the banana region through HSV color space conversion, Gaussian smoothing, Otsu thresholding, and morphological cleanup to generate a reliable binary mask. Ripeness is then estimated by analyzing only banana pixels, extracting simple color-based features such as green and yellow proportions from the Hue channel and dark-spot proportions from the Value channel. These features are used in a threshold-based rule classifier to categorize bananas as unripe, ripe, or overripe.

Since the dataset was not fully labeled, a numerical accuracy metric could not be computed. However, manual inspection across multiple samples demonstrated that the method performs reliably under the intended setup, where each image contains a single banana with a mostly plain background.

For future improvements, the system can be extended to support images containing multiple bananas and to operate reliably under more complex backgrounds with multiple objects or highly saturated colors. Additionally, the current HSV-based thresholds are sensitive to lighting conditions and camera angles. Alternative color representations, such as the Lab color space, or adaptive learning-based methods could be explored to further improve robustness and accuracy.

References

- Mendeley Data. *Banana Dataset (yellow and dark banana images).* `https://data.mendeley.com/datasets/ptfscwtnyz/2`
- Kaggle. *Banana Ripeness Classification Dataset (green banana images).* `https://www.kaggle.com/datasets/shahriar26s/banana-ripeness-classification-dataset/data`