Color-Based Fruit Detection Using Classical Image Processing Techniques

Project Description

An Image Processing Project on Color Segmentation and Object Detection

Project Members

Arvind Mahalingam
Dipti Ramnath
Sujitha C.B

Abstract

Automatic fruit detection is an important task in smart agriculture, food quality inspection, and automated retail systems. This project presents a classical image processing-based approach for detecting fruits using color information without employing machine learning or deep learning techniques.

The proposed system utilizes HSV color space conversion, threshold-based segmentation, morphological filtering, and connected component analysis to accurately identify and localize fruit regions in images. Experimental evaluation is carried out using fruit images from the Fruits-360 dataset. The results indicate that the proposed method provides reliable detection for fruits with distinct color characteristics while maintaining low computational complexity, making it suitable for real-time and resource-constrained environments.

1 Introduction

Automation in agriculture and food processing industries has created a growing demand for efficient and cost-effective fruit recognition systems. Applications such as automated fruit sorting, grading, inventory management in supermarkets, and quality inspection require accurate identification of fruits with minimal human intervention.

While machine learning and deep learning techniques often achieve high accuracy, they require large labeled datasets and significant computational resources. In contrast, classical image processing techniques offer a simpler and more efficient alternative for environments with limited resources.

The key contributions of this project are:

- Design of a non-learning-based fruit detection pipeline
- Use of HSV color space for robust color segmentation
- Application of morphological filtering and connected component analysis

2 Literature Survey

Early fruit detection approaches relied on RGB color thresholding, which is highly sensitive to illumination variations. To overcome this limitation, alternative color spaces such as HSV and LAB were introduced, which separate chromatic information from intensity.

Morphological operations have been widely used to refine segmentation masks. Although deep learning-based approaches dominate recent research, classical image processing techniques remain relevant due to their simplicity, interpretability, and low computational cost.

3 Proposed Enhancement Over Existing Works

The proposed system relies entirely on deterministic image processing techniques, eliminating the need for training and large datasets. By combining HSV-based segmentation with morphological filtering and connected component analysis, consistent fruit detection is achieved.

The modular design allows easy extension to other fruit types by modifying color thresholds, making the system suitable for real-time and low-resource applications.

4 Proposed Methodology

The fruit detection system follows the pipeline shown below:

Input Image → HSV Conversion → Color Thresholding → Morphological Filtering → Region Labeling

4.1 Color Space Conversion

The RGB image is converted into HSV color space to improve robustness against illumination variations.

4.2 Thresholding

Fixed hue and saturation thresholds are selected to isolate fruit regions based on color characteristics.

4.3 Morphological Filtering

Opening and closing operations are applied to remove noise and fill small gaps in the segmented mask.

4.4 Region Labeling

Connected component analysis is used to localize fruit regions, and bounding boxes are drawn around detected objects.

5 Implementation

5.1 Fruit Detection Algorithm

```python
import cv2
import numpy as np

img = cv2.imread("images/banana.jpg")
if img is None:
    print("Error: Image not found")
    exit()

hsv = cv2.cvtColor(img, cv2.COLOR_BGR2HSV)

lower_yellow = np.array([20, 100, 100])
upper_yellow = np.array([30, 255, 255])
mask = cv2.inRange(hsv, lower_yellow, upper_yellow)

kernel = np.ones((5, 5), np.uint8)
mask = cv2.morphologyEx(mask, cv2.MORPH_OPEN, kernel)
mask = cv2.morphologyEx(mask, cv2.MORPH_CLOSE, kernel)

num_labels, labels, stats, centroids = cv2.connectedComponentsWithStats(mask)

for i in range(1, num_labels):
    if stats[i, cv2.CC_STAT_AREA] > 1000:
        x, y, w, h, _ = stats[i]
        cv2.rectangle(img, (x, y), (x+w, y+h), (0,255,0), 2)
        cv2.putText(img, "Banana", (x, y-10),
                    cv2.FONT_HERSHEY_SIMPLEX, 0.8, (0,255,0), 2)

cv2.imshow("Detected Fruit", img)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Listing 1: Color-Based Fruit Detection Using HSV Segmentation

6 Results and Discussion

The proposed system demonstrated reliable segmentation and localization for fruits with distinct color characteristics such as bananas and apples. The use of HSV color space significantly improved robustness compared to RGB-based approaches.

6.1 Visual Detection Results

Figure 1 presents the visual output of the proposed fruit detection system, including the original image, segmentation mask, and final detected result with bounding box and label.

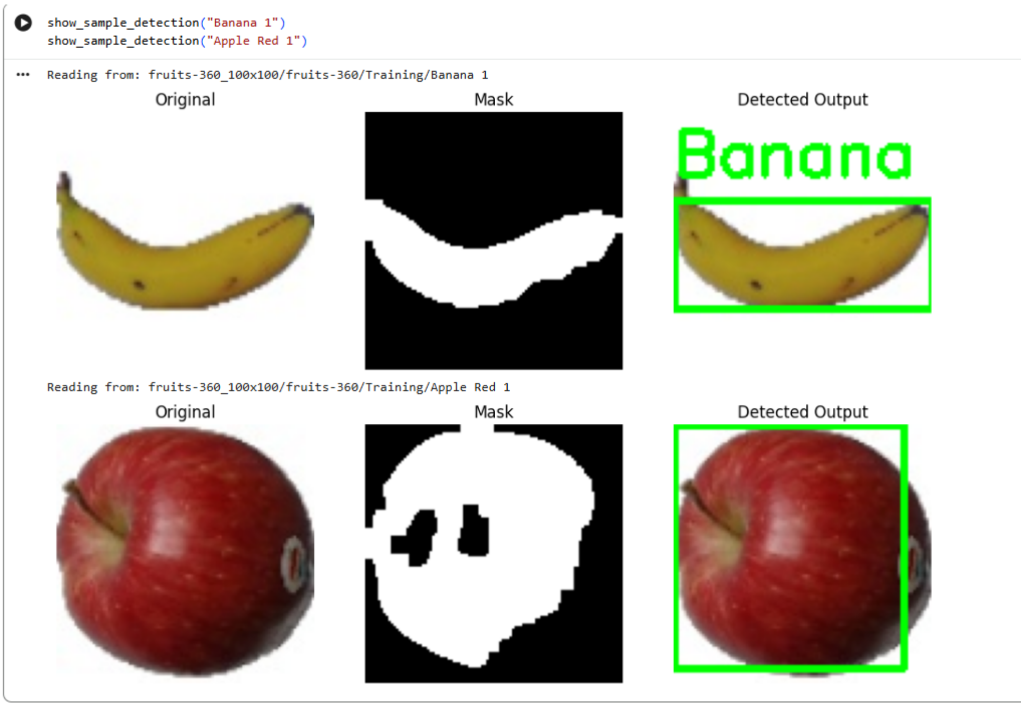

Figure 1: Visual results showing original image, segmentation mask, and detected fruit

7 Conclusion and Future Scope

This project successfully demonstrated a color-based fruit detection system using classical image processing techniques. Accurate fruit localization was achieved without machine learning by leveraging HSV color space and morphological filtering.

Future work may include adaptive thresholding, illumination normalization, multi-fruit detection, and real-time video processing.

References

[1] Gonzalez, R. C., and Woods, R. E., *Digital Image Processing*, Pearson Education.

[2] H. Muresan and M. Oltean, “Fruit recognition from images using deep learning,” Acta Univ. Sapientiae, 2017.