Fruit Freshness Detection Algorithm Documentation

1 Overview

The fruit freshness detection algorithm employs a multi-feature analysis approach to determine the ripeness of fruits. The three primary features used are:

- **Shape Analysis:** Analyzing the geometric properties of the fruit.
- **Color Analysis:** Evaluating the color characteristics to determine freshness.
- **Texture Analysis:** Identifying surface textures of the fruit to classify freshness.
- **Dark Spots Detection:** Identifying any dark spots or bruising, which are indicators of overripe or rotten fruits.

The algorithm uses pre-trained machine learning models to identify and segment the fruit in images. Once the fruit is detected, various properties (shape, color, and texture) are analyzed, and a reward function is employed to compute a freshness score. The final freshness score is then used to classify the fruit as either "fresh," "moderately stale," or "rotten."

2 Algorithm Flow

2.1 Input Image Processing

- The input image is first pre-processed to extract the region of interest (ROI) containing the fruit.
- A deep learning-based object detection model, like YOLO, is used to detect and segment the fruit in the image.

2.2 Shape Analysis

- The algorithm extracts the shape of the fruit by analyzing its contours.

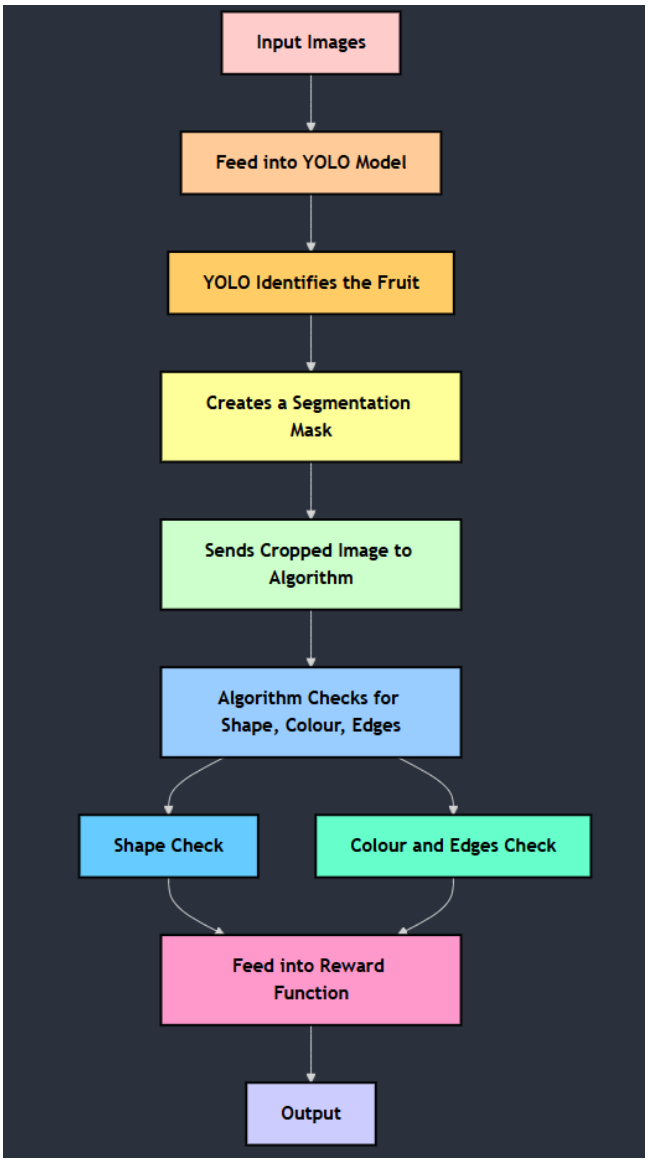

Figure 1: Flowchart of the Algorithm.

- Common geometric properties such as circularity, aspect ratio, and compactness are computed to determine if the fruit's shape matches that of a fresh fruit.

For example:

- A fresh apple might have a circular or slightly oval shape, while a rotten

apple may have distorted edges due to decomposition.

2.3 Color Analysis

- The color of the fruit is analyzed by converting the image from RGB to HSV (Hue, Saturation, Value) space.
- Key color features such as hue, saturation, and brightness are extracted.
- Fresh fruits typically have vibrant colors, high saturation, and moderate brightness.
- Rotten fruits may have dull or darkened hues with low saturation.

Color thresholds are set for each fruit type to detect freshness:

- For example, the hue of an apple may be between 0°-30° for freshness, with a high saturation value (80-100), and brightness (Value) between 50-100.

2.4 Texture Analysis

- The texture of the fruit is assessed by applying texture analysis techniques, such as Local Binary Patterns (LBP) or Gabor Filters.
- Texture analysis helps identify surface roughness, spots, and bruising, which are characteristic of ripe or overripe fruits.
- Fresh fruits generally have smooth, even surfaces, while rotten fruits may show signs of decay such as wrinkles, spots, or mold.

2.5 Dark Spots Detection

- Dark spots are often a sign of bruising, over-ripeness, or decay, which are common in overripe or rotten fruits.
- The algorithm uses image processing techniques to detect dark spots or areas with low brightness (dark pixels), which are typically darker than the surrounding regions.
- Methods like thresholding or blob detection are applied to isolate and highlight dark areas on the fruit's surface.
- A high number of dark spots or large dark areas in the fruit indicates that the fruit is likely to be rotten or stale.

2.6 Reward Function

After extracting the features (shape, color, and texture), a reward function is used to calculate the freshness score. The reward function takes the extracted values and compares them against predefined threshold values for each feature. Each feature contributes to the overall freshness score, which is calculated as follows:

$$\text{Freshness Score} = w_1 \times \text{Shape Score} + w_2 \times \text{Color Score} + w_3 \times \text{Texture Score} \quad (1)$$

Where:

- w_1, w_2, w_3 are weights assigned to each feature based on its importance.
- The score for each feature is normalized between 0 and 1.

2.7 Freshness Classification

The output of the reward function is mapped to a class label (e.g., "Fresh," "Moderately Stale," or "Rotten"). The classification is used to make recommendations about the fruit, such as "Eat now," "Consume soon," or "Discard."

3 Libraries Used

The following libraries are utilized in the fruit freshness detection pipeline:

- **OpenCV (cv2):** Used for image processing, such as resizing, color space conversion (RGB to HSV), contour detection, and texture analysis.
- **Ultralytics YOLO:** Used for object detection to identify and segment the fruit in the input image. YOLO helps detect the bounding box and class labels for the fruit.
- **NumPy:** Used for numerical computations, such as calculating shape features (aspect ratio, circularity) and implementing the reward function.
- **Scikit-Image:** Used for texture analysis, including methods like Local Binary Patterns (LBP) for surface analysis.

4 Feature Extraction Details

4.1 Shape Features

- **Circularity:** Measures how close the shape is to a perfect circle. A fresh fruit will have a near-perfect circularity.
- **Circularity Formula:**

$$\text{Circularity} = \frac{4 \times \pi \times \text{Area}}{\text{Perimeter}^2}$$

- **Aspect Ratio:** Measures the ratio of width to height of the bounding box surrounding the fruit. Fresh fruits tend to have a balanced aspect ratio.

4.2 Color Features

- **Hue:** Represents the color of the fruit. Fresh fruits have vibrant hues within specific ranges (e.g., for apples, 0°-30°).
- **Saturation:** Represents the intensity of the color. Fresh fruits have high saturation, while rotten fruits have low saturation.
- **Brightness (Value):** Represents the lightness or darkness of the color. Fresh fruits have a moderate brightness value.

4.3 Texture Features

- **Local Binary Patterns (LBP):** Describes the texture based on pixel intensity. Smooth textures are typical of fresh fruits.
- **Gabor Filters:** Used to detect texture patterns such as roughness, spots, or bruising.

5 Threshold Values

Thresholds are calculated as the mean of training data.

5.1 Shape:

- **Fresh:** Circularity > 0.8, Aspect Ratio close to 1.
- **Rotten:** Circularity < 0.5, Aspect Ratio significantly deviating from 1.

5.2 Color:

- **Fresh:** Hue between 0°-30° (for apples), Saturation $> 70\%$, Brightness between 50-100.
- **Rotten:** Hue outside typical ranges, Saturation $< 40\%$, Brightness < 50.

5.3 Texture:

- **Fresh:** Low roughness, smooth texture.
- **Rotten:** High roughness, visible bruises or mold.

6 Features of the Algorithm

This algorithm is optimized for real-time performance, utilizing basic image processing techniques like shape, color, and texture analysis. By avoiding complex models or external API calls, it ensures fast processing and immediate results, making it ideal for applications requiring quick feedback.

7 Example of How Different Features Play a Role:

- **Fresh Fruit Example:** This is a fresh fruit, so it is circular in shape, has a vibrant red color. All these features give it a high reward value making it fresh.

Figure 2: Fresh fruit example showing circular shape and vibrant color.

- **Rotten Fruit Example:** This apple is rotten, which is evident from the fact that it is wrinkled and has dark spots. But it still maintains its shape. So that shape factor cannot detect alone that it is rotten, but combined with the factors for dark spot detection and edge detection, that can detect wrinkles, we can make the correct prediction.

Figure 3: Rotten fruit example showing wrinkles and dark spots.