Objective:

- **To develop** an automated system for identifying fruits in digital images.
- **To be familiar with** the color properties of different fruits using color-based segmentation and minimum distance classification.

Description:

This project aims to develop a user-friendly system for identifying fruits in digital images. The system will display the original image, the segmented fruit region, and the identified fruit name with confidence information. The success of the project will be evaluated based on the accuracy of fruit identification compared to ground truth labels for a set of test images. The initial implementation will focus on identifying a set of common fruits with distinct color characteristics. This is a starting point for fruit identification, focusing on color and targeting common fruits. Overall, this project aims to provide a basic yet functional tool for fruit identification based on color analysis, promoting user interaction and basic fruit color recognition.

Formulation of This Problem:

As we know, complex engineering problems have some characteristics of P1 to P7. For complex engineering problems, there must be touched P1 and one or more from P2 to P7.
In This project it has touched P1, P2 and P3.

P1 (Depth of Knowledge Required):
To find fruit by its color, a profound understanding of color classification which is engineering fundamentals (K3) is necessary. A depth of knowledge is needed for precise color analysis and threshold definition. Understanding the specific color characteristics of different fruits (K4) is necessary.

P2 (Range of Conflicting Requirements):
In a image there can be diverse colors, lighting conditions can be different. which is so much of a conflict between color and fruits. Striking a balance among these conflicting requirements (P2) is a complex challenge.

P3 (Depth of Analysis Required):
Developing a Fruit detector demands a comprehensive analysis of different type and different value. color data.
For this reason, Leaf Health Assessment is a Complex Engineering.

Methodology:

Data Definition:

- A list of fruits is defined with example names like Apple, Banana, etc.
- Corresponding average Hue and Saturation values are stored(predefined) for each fruit. These values obtained manually analyze the average Hue and Saturation of an image in MATLAB.

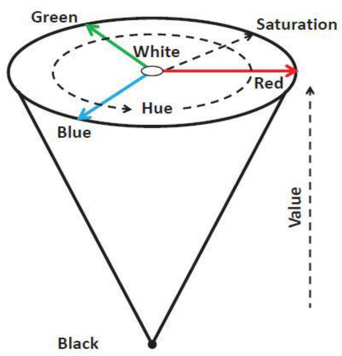
User Input:

- We used a dialog box allowing the user to select an image file (jpg, png, bmp).

It checks if a file was selected to proceed further Image Processing:

Color Space Conversion (RGB to HSV):

- The image is then converted from the RGB color space to the HSV color space using `rgb2hsv`. HSV is often preferred for color-based analysis due to its separation of hue, saturation (color intensity).



In the HSV space the color model is represented with the components hue (H), saturation (S) and value (V) [1]

- Hue is the chromatic feature that describes a pure color, for instance, yellow, orange, red, etc.
- Saturation is a measure of how the hue is diluted in white light.
- value is the intensity or brightness of the color.

The hue and saturation components emulate the human perception of color. With these features the HSV space becomes a useful tool to develop image processing algorithms based on some properties of human color perception.

Feature Extraction:

- calculates the average Hue and average Saturation of the entire image using `mean2`. This essentially provides a single representative color value based on hue and saturation for the whole fruit in the image.

Fruit Identification:

- For each fruit, the Euclidean distance between the image's average Hue and Saturation and the stored average value for that particular fruit is calculated.

Euclidean distance refers to the straight-line distance between two points in Euclidean space. it's the shortest distance traveling between two points in a straight line. Imagine stretching a taut string between the two points - that's the Euclidean distance.[2]

Mathematical equation:
Dist = sqrt ((x2 - x1) ^2 + (y2 - y1) ^2)

It measures the "straight-line" distance between two points in a multi-dimensional space (in this case, a 2D space of Hue and Saturation).

If the calculated distance for the current fruit type is less than the current minimum distance, it means we found a closer match to the stored fruit.

Result Display:

- Display the original image , HSV image and output image .The assessment outcome based on the analysis of the HSV value for the specific fruit detection.

Results:

When image is selected:

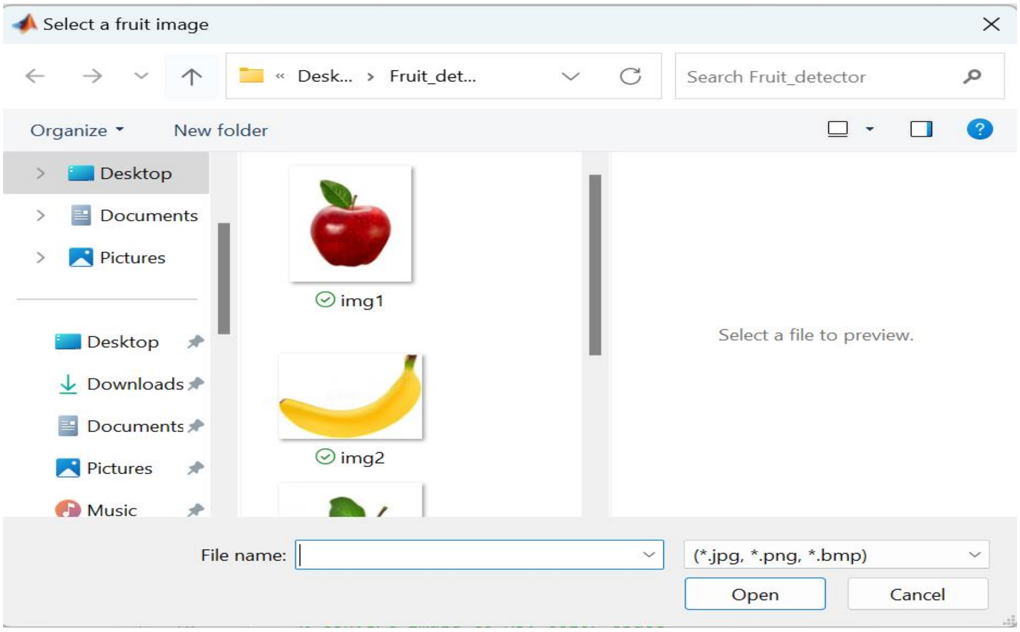


Figure 1 Taking Input image from dialog box.

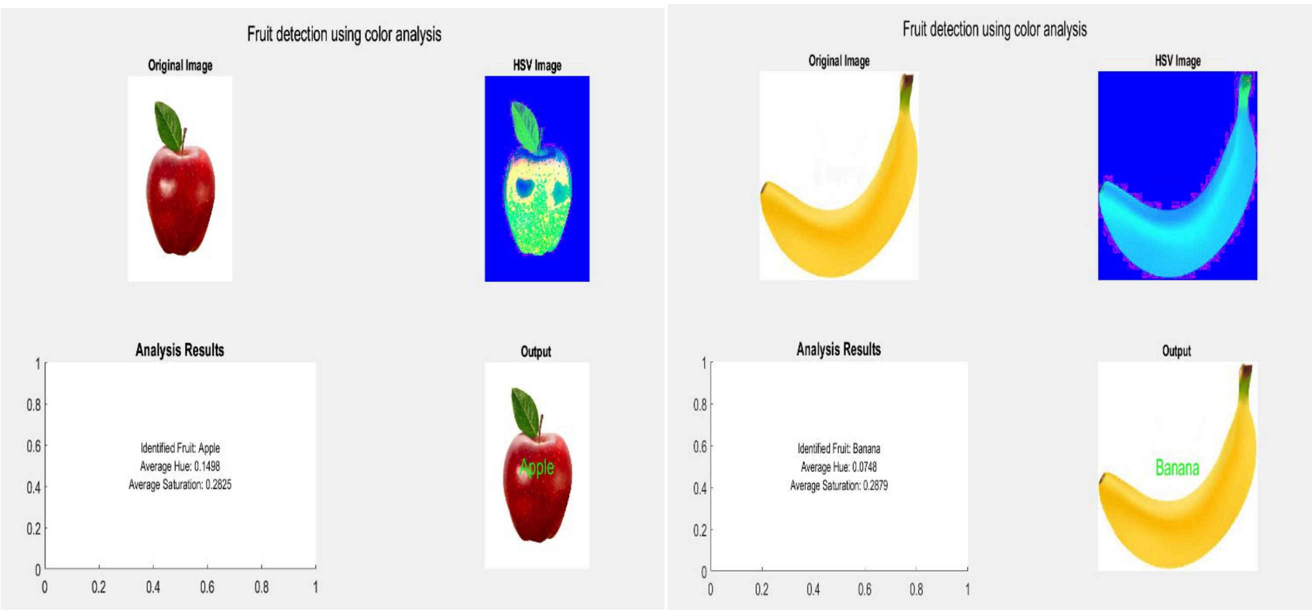


Figure 2 Output image and analysis of fruit detection.

when no image is selected:

```
>> fruit_main_detector
No image selected.
```

Figure 3: Output will be no image is selected.

In this project we have given an input image (imread) through dialog box and then processed image to find out the accurate fruit from the image. in figure 2 we can see the output and analysis report from this project. Our project have run successfully, and the result is successful.

Conclusion:

This project successfully developed a user-friendly system for basic fruit identification in digital images. It leverages color analysis through Color-Based Segmentation, Feature Extraction and Minimum Distance Classification. The current implementation focuses on a limited set of common fruits with distinct color properties. It calculates average Hue and Saturation values to represent the dominant color characteristics of the fruit. The minimum distance classification offers a basic approach. The fruit with the features closest (minimum Euclidean distance) to the extracted values is considered the most likely match. The system displays the original image and the identified fruit name, providing a clear result for the user.

There are some potential areas for future work in this project:

Fruit detection by

- Advanced Segmentation Techniques like Machine Learning-Based Segmentation and edge detection.
- Feature Extraction by shape analysis.
- Machine Learning Classification like KNN and CNN method.
- Real-Time Fruit Detection.

Appendix:

Code	Explanation
fruits = {'Apple', 'Banana', 'Guava', 'Strawberry'};	This line defines a cell array named fruits. A cell array can hold different data types in a single variable. Here, it stores strings representing known fruit names: "Apple", "Banana", "Guava", and "Strawberry".
fruit_hues = [0.1497, 0.07481, 0.0658, 0.1243]; **fruit_sats = [0.2824, 0.2878, 0.1803, 0.1631];**	These lines define two numeric arrays: fruit_hues and fruit_sats. There are pre calculated values for hue and saturation for each corresponding fruit name in fruits (same order).
[filename, pathname] = uigetfile({'*.jpg;*.png;*.bmp'}, 'Select a fruit image');	This line displays a file selection dialog box using uigetfile, the functions takes two arguments. **{'*.jpg; *.png;*.bmp'}:** This filters the dialog box to only allow image files with JPG, PNG, or BMP extensions.
if ~isequal(filename, 0)	This line checks if a file was selected using uigetfile,

	~isequal compares two values and returns true if they are not equal.Here, it checks if filename is not equal to 0
path = fullfile(pathname, filename); **I = imread(path);**	These lines are only executed if a file was selected, the full path to the image file by combining the directory path (pathname) and filename (filename) using **fullfile**. the image data from the specified path using **imread.** The output is stored in a variable named I.
HSV = rgb2hsv(I);	This line converts the image data (I) from RGB (Red, Green, Blue) color space to HSV (Hue, Saturation, Value) color space. The **rgb2hsv** function performs this conversion.
avg_hue = mean2(HSV (: ,:, 1)); **avg_sat = mean2(HSV(:, :, 2));**	These lines calculate the average Hue and Saturation values for the entire image. The **mean2** function calculates the element-wise mean of a matrix. Here, it's applied to the first two channels of the HSV image HSV(:,:,1) for Hue and HSV(:,:,2) for Saturation to get their average values.
min_dist = inf; **fruit_idx = -1;** **for i = 1:length(fruits)**	min_dist is assigned the value **inf**, which represents positive infinity in MATLAB. fruit_idx is assigned the value **-1** Next line starts a **for** loop. The loop variable i will iterate from 1 to the length of the fruits cell array. In each iteration, i will represent the index of a particular fruit in the reference data.

dist = sqrt((avg_hue - fruit_hues(i))^2 + (avg_sat - fruit_sats(i))^2);
if dist < min_dist
    min_dist = dist;
    fruit_idx = i;
end
	This code calculates the Euclidean distance between the average Hue **(avg_hue)** and Saturation **(avg_sat)** of the image and the corresponding average values for the current fruit **fruit_hues(i)** and **fruit_sats(i)** stored in the reference data.
	The **sqrt** function calculates the square root.
	This distance represents how similar the image's color characteristics are to the current fruit in the reference data.
	if dist < min_dist
	This if statement checks if the calculated distance (dist) is less than the current minimum distance (min_dist).
	min_dist = dist; If the calculated distance is smaller, this line updates min_dist to hold the new minimum distance.
	fruit_idx = i; If the distance is smaller, this line updates fruit_idx to store the current index (i), indicating that this fruit is a closer match.

if fruit_idx == -1 **frt_name = 'Unknown Fruit';** **else**	This is an **if-else statement** that checks the value of **fruit_idx**. If **fruit_idx** is still equal to -1 (meaning no fruit with a close enough distance was found

frt_name = fruits{fruit_idx}; **end**	in the loop), this condition is true.The code assigns the string "Unknown Fruit" to the variable frt_name. This indicates that the image's color characteristics didn't match any fruit closely enough in the reference data. If **fruit_idx** is not equal to -1 (meaning the loop found a fruit with a minimum distance), this block is executed. **frt_name = fruits{fruit_idx};** retrieves the name of the fruit from the fruits cell array using the index stored in fruit_idx. fruits is a cell array containing fruit names as strings.By using curly braces {} after fruits, the code accesses the element at the specific index (fruit_idx) within the cell array.
figure; **subplot(2,2,1);** **imshow(I);** **title('Original Image');**	Showing the original image in subplot (221)
subplot(2,2,2); **imshow(HSV);** **title('HSV Image');**	Showing the HSV in subplot (222)
subplot(2,2,3, 'FontSize', 12); **text(0.5, 0.5, sprintf('Identified Fruit: %s\nAverage Hue: %.4f\nAverage Saturation: %.4f', ... frt_name, avg_hue, avg_sat), ... 'HorizontalAlignment','center','VerticalAlignment', 'middle');** **title('Analysis Results');**	Showing the Analysis in subplot (223) **text (0.5, 0.5, ...):** This defines the position of the text within the subplot. (0.5, 0.5) represents the center of the subplot (normalized coordinates). **sprintf(...):** This function formats a string with placeholders. Here, it creates a string containing data which shown in the analysis plot.
subplot(2,2,4); **imshow(I);** **title('Output ');**	Showing the output in subplot (224), if next line (if strcmp) is true.
if strcmp(frt_name, 'Unknown') **text(size(I,2)/2, size(I,1)/2, 'Unknown Fruit', ...** **'HorizontalAlignment', 'center', 'VerticalAlignment', 'middle', ...** **'FontSize', 16, 'Color', 'red');** **else** **text(size(I,2)/2, size(I,1)/2, frt_name, ...** **'HorizontalAlignment', 'center', 'VerticalAlignment', 'middle', ...** **'FontSize', 16, 'Color', 'green');** **end**	This code add text over the image in the subplot, depending on the identified fruit if **strcmp(frt_name, 'Unknown')** checks if frt_name is equal to "Unknown" **(using strcmp for string comparison).**If the condition is true, the code inside the if block executes. Else If frt_name is not "Unknown", the code in the **else** block executes.
sgtitle('Fruit detection using color analysis');	Title will be Fruit detection using color analysis
fprintf('\nThe given fruit is: %s\n\n', frt_name); **else** **fprintf('No image selected.\n');** **end**	**fprintf** will print the fruit name of the given fruit else no image will be selected.

Reference:

[1] F. Garcia-Lamont, J. Cervantes, A. López, and L. Rodriguez, “Segmentation of images by color features: A survey,” *Neurocomputing*, vol. 292, pp. 1–27, May 2018, doi: 10.1016/j.neucom.2018.01.091.

[2] L. Ganesan, S. Arivazhagan, R. N. Shebiah, S. Selva Nidhyanandhan, and L. Ganesan, “Fruit Recognition using Color and Texture Features,” 2010. [Online]. Available: http://www.cisjournal.org