Khulna University of Engineering & Technology
Department of Computer Science and Engineering

CSE 4128: Image Processing and Computer Vision Laboratory

Project Title: **Flora (Living Flowers) Counting**

Instructed by:
Dr. Sk. Md. Masudul Ahsan
Professor,
Department of CSE, KUET

Instructed by:
Dipannita Biswas
Lecturer,
Department of CSE, KUET

Submitted by:

Nushrat Tarmin Meem

Roll: **1907083**

Group: B_2

Year: 4^{th}

Semester: 1^{st}

Date of Submission: 01.07.2024

Flora (Living Flowers) Counting

Flora is a term that means "Goddess of the Flower" in Latin which is a collective term for a group of plant life found in a particular region. The whole plant kingdom is represented by this name. In Roman mythology, Flora was the goddess of spring and flowering plants, especially wildflowers and plants not raised for food. Specifically, flora indicates the flowers that is basically staying on living plants.

Objectives:

- Developing a system to automatically count the number of flowers in an image
- Improving accuracy compared to manual counting methods
- Providing an easy-to-use interface for uploading images and viewing results
- Analyzing and processing images to detect and count flowers
- Enabling real-time flower counting for field applications

Introduction:

Counting flowers in images involves using image processing techniques such as noise reduction, thresholding, and segmentation to isolate flowers from backgrounds. Features like color and shape are extracted using methods like contour detection. Classification algorithms are then employed to distinguish flowers from non-flowers, facilitating accurate counting. Challenges include variability in flower types and environmental conditions, requiring robust validation and iterative refinement to enhance algorithm accuracy and applicability across diverse datasets and real-world scenarios.

Purpose of this project: Automating flower counting can help botanists, farmers, and researchers efficiently monitor flower growth and health

Input: Images of different colored flowers

Output: An annotated image showing detected flowers with numbers

Functionalities:

Different techniques involving -

- User Friendly Environment: Allowing users to upload images of flowers
- Preprocessing: Converting images to a format suitable for processing (e.g., resizing, converting to grayscale or HSV).
- Clustering: Using k-shift clustering and showing image with the main colors only
- Color Filtering: Using color ranges to isolate flowers from the background
- Contour Detection: Identify flower contours using edge detection methods
- Counting: Counting the number of detected flower contours
- Annotation: Annotating the original image with detected flower boundaries and displaying the count with various colored numbers
- User Interface: Providing a graphical interface for interaction

Working Process:

❑ User Interface:

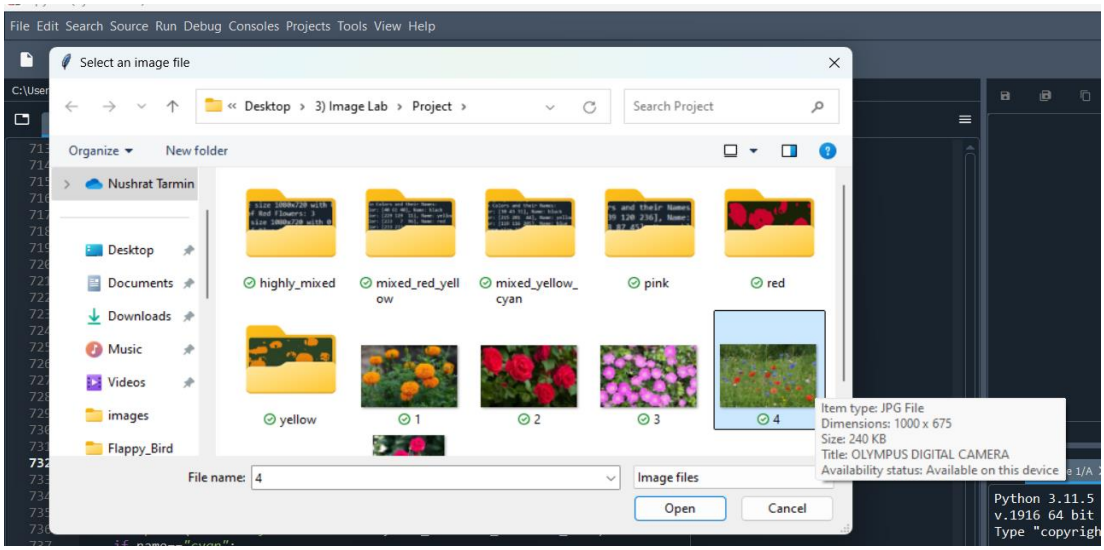


Figure 1: User friendly interface

✓ Providing a way to upload images
✓ Displaying the processed image with annotations and the count of flowers

❑ Image Preprocessing:

✓ Converting the input image to HSV color space to make color filtering easier
✓ Applying Gaussian blur to reduce noise and improve contour detection.

❑ Mean-shift Clustering:

K-shift clustering is a variant of the K-means clustering algorithm that seeks to improve clustering performance by dynamically adjusting cluster centroids based on the density of data points. Unlike K-means, which initializes centroids randomly and iteratively assigns points to the nearest centroid, K-shift starts with a fixed number of centroids (K) and then adjusts their positions iteratively to better capture the density peaks of the data. This adaptation allows K-shift to handle clusters of varying shapes and densities more effectively than traditional K-means, making it suitable for datasets with complex structures

✓ Initializing K centroids in the color space
✓ Iterating through the pixels of the image, assigning each pixel to the nearest centroid based on color similarity
✓ Adjusting centroids based on the density of nearby pixels to better represent clusters of colors

❑ **Identifying Main Colors:**

✓ Once clustering is complete, each centroid represents a cluster of similar colors

✓ Extracting the centroids (cluster centers) as the main representative colors found in the image

```
# Manual dictionary of basic colors
BASIC_COLORS = {
    'black': '#000000',
    'white': '#FFFFFF',
    'red': '#FF0000',
    'green': '#00FF00',
    'blue': '#0000FF',
    'yellow': '#FFFF00',
    'cyan': '#00FFFF',
    'magenta': '#FF00FF'
}
```

Figure 2: Web colors used for clustering

❑ **Color Filtering**:

✓ Defining color ranges for flower detection (e.g., red, yellow, blue, cyan, pink, magenta, white etc.)

✓ Creating a mask to filter out everything except the desired color range

❑ **Color Rejection:**

If the segmented color region is <= 1% or (green || black) portion is >= 99% then that color of flower is not going to be counted

❑ **Canny Edge Detection:**

For finding thin edges, canny edge detection algorithm has been used.

❑ **Contour Detection**:

✓ Using OpenCV's find Contours function to detect edges and boundaries of flowers

✓ Filtering out small contours that are likely noise

❑ **Counting**:

Counting the number of filtered contours which represent individual flowers.

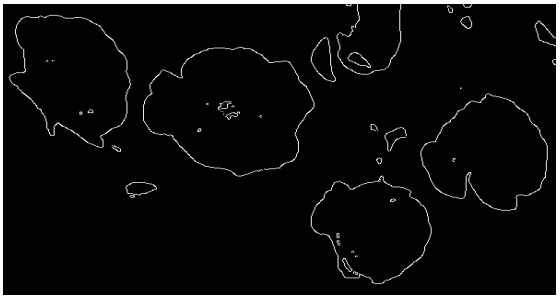
❑ **Annotation**:

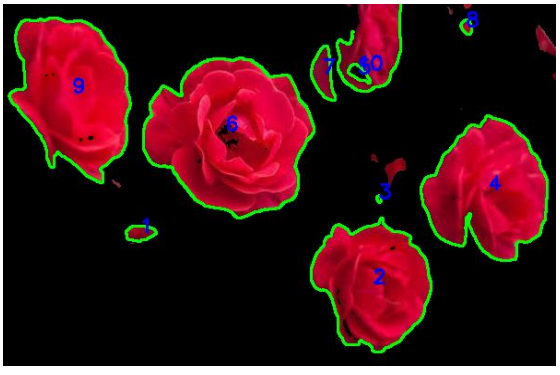
✓ Drawing bounding boxes or contours around detected flowers on the original image

✓ Displaying the count on the image

Result of Working Process:

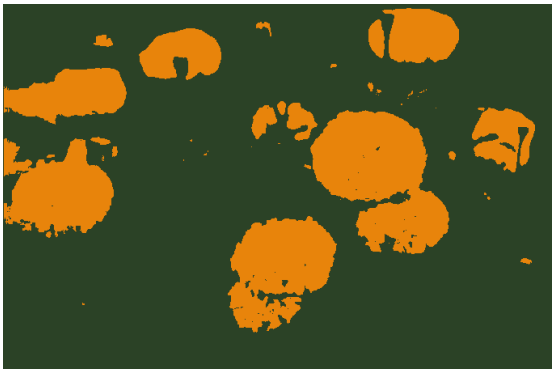
```
Main Colors and their Names:
Color: [26 35  7], Name: black
Color: [200   3  43], Name: red
<Figure size 1080x720 with 0 Axes>
Number of Red Flowers: 10
```

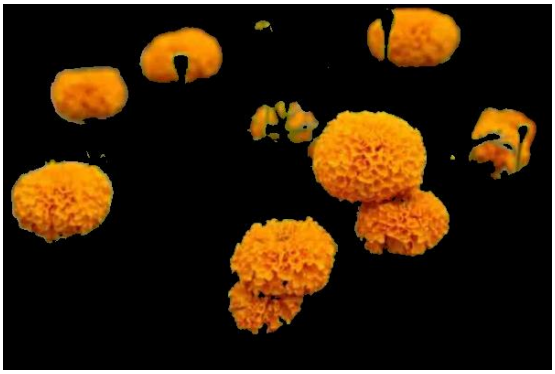


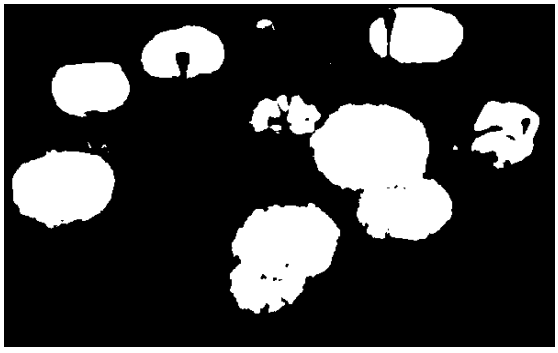


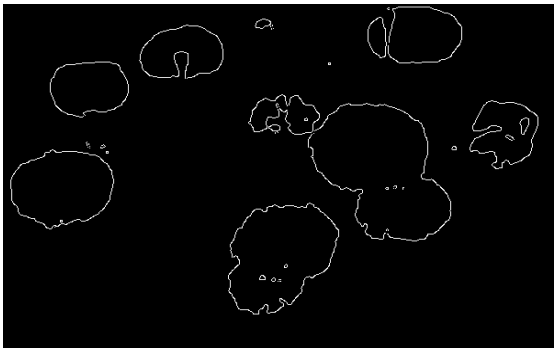
Result of Working Process:

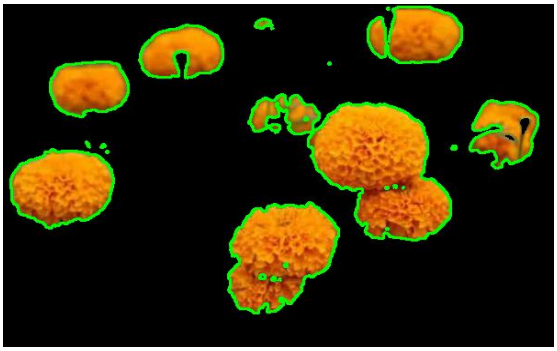

```
Main Colors and their Names:
Color: [43 66 38], Name: black
Color: [232 132  11], Name: yellow
<Figure size 1080x720 with 0 Axes>
Number of Yellow Flowers: 12
```

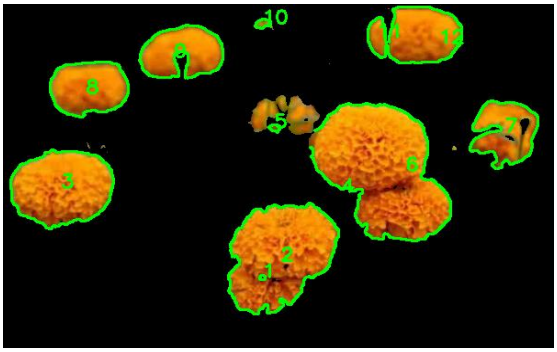






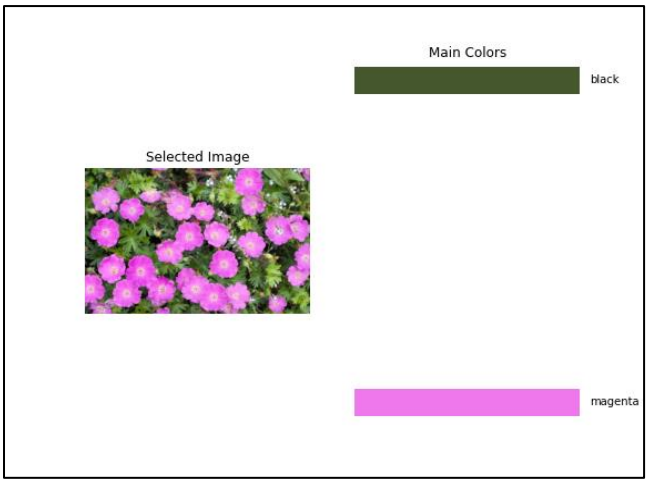


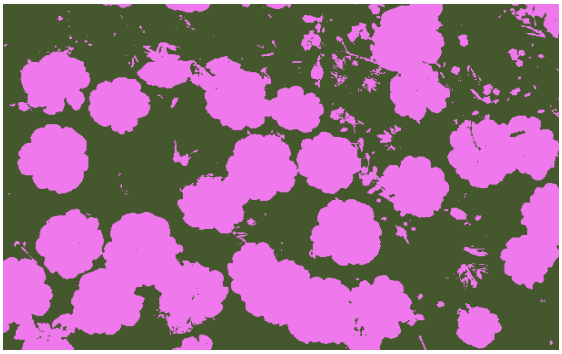


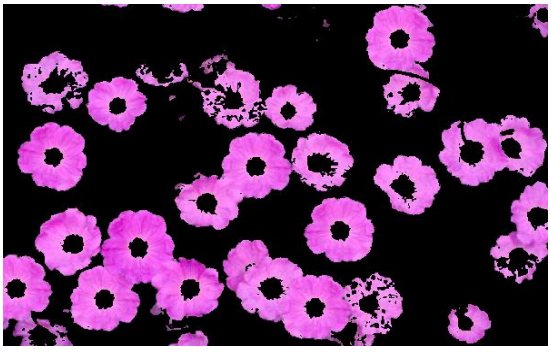


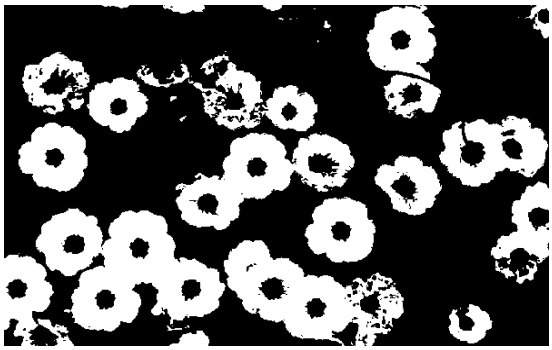
Result of Working Process:

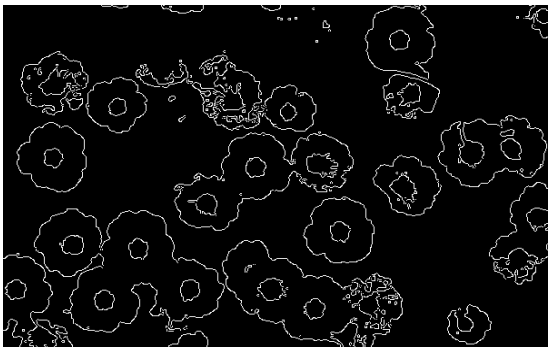
```
Main Colors and their Names:
Color: [239 120 236], Name: magenta
Color: [68 87 45], Name: black
<Figure size 1080x720 with 0 Axes>
Number of Magenta Flowers: 25
```

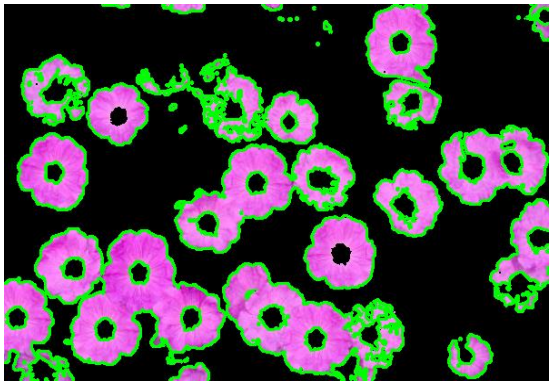


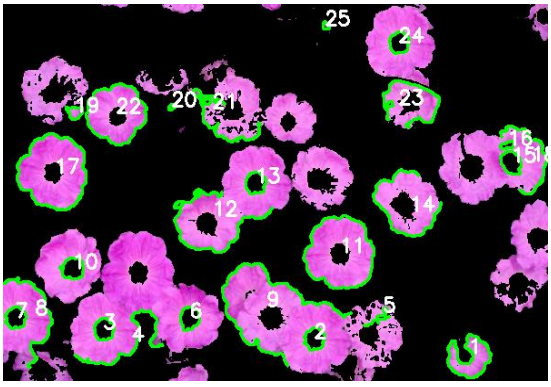




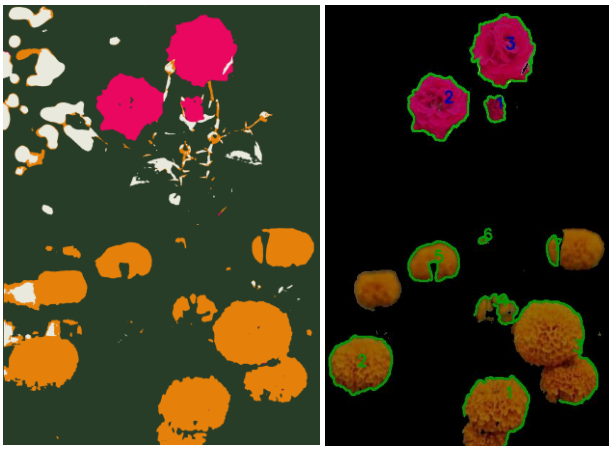


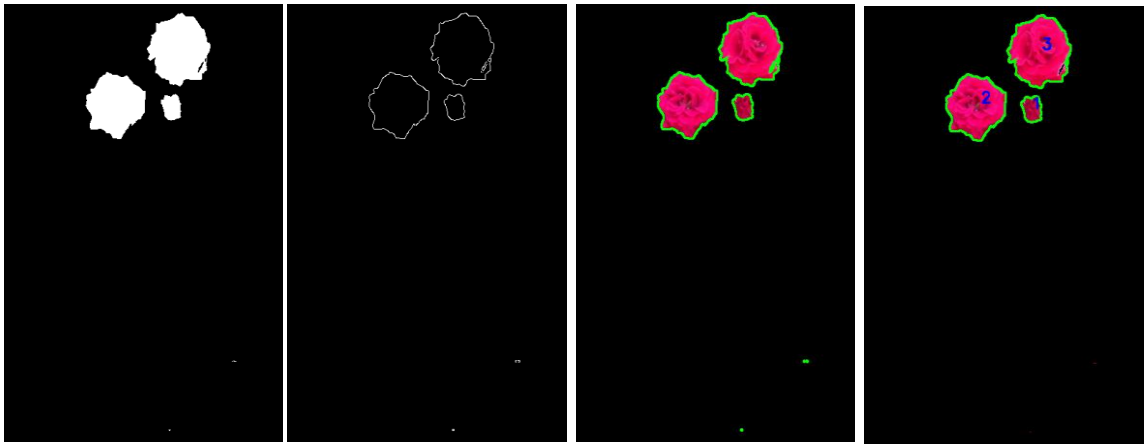


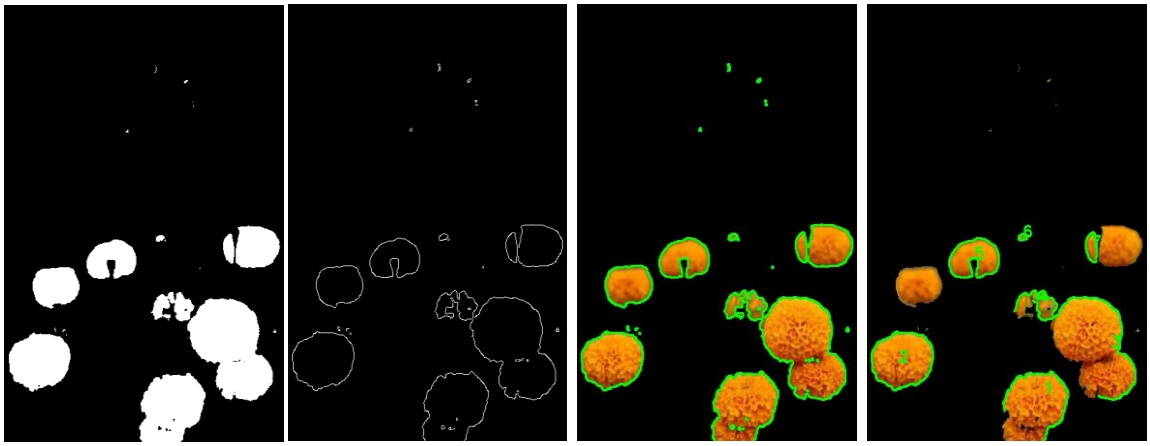




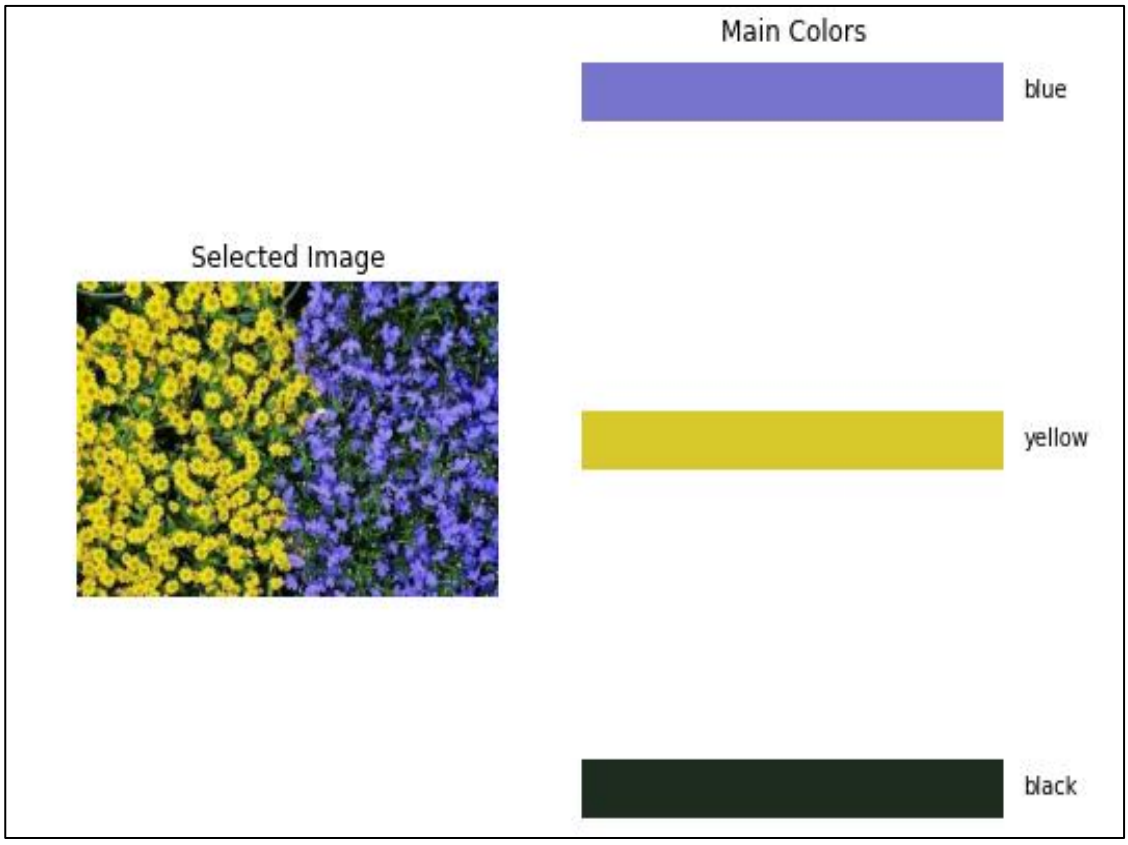
Result of Working Process:

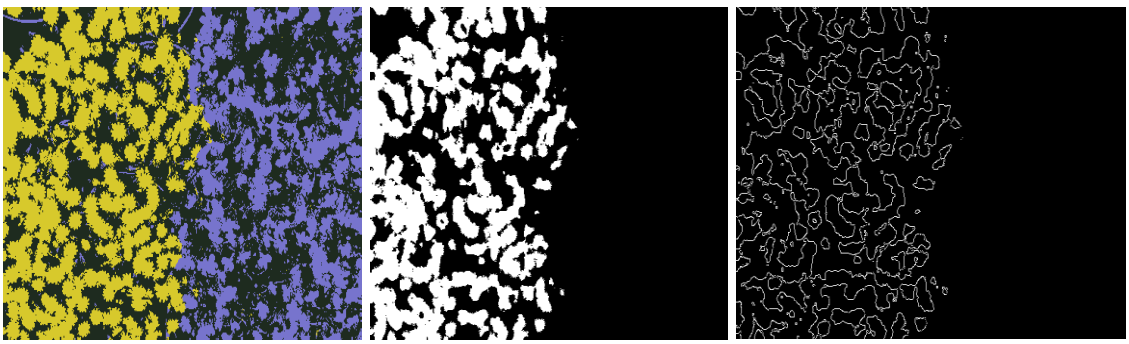


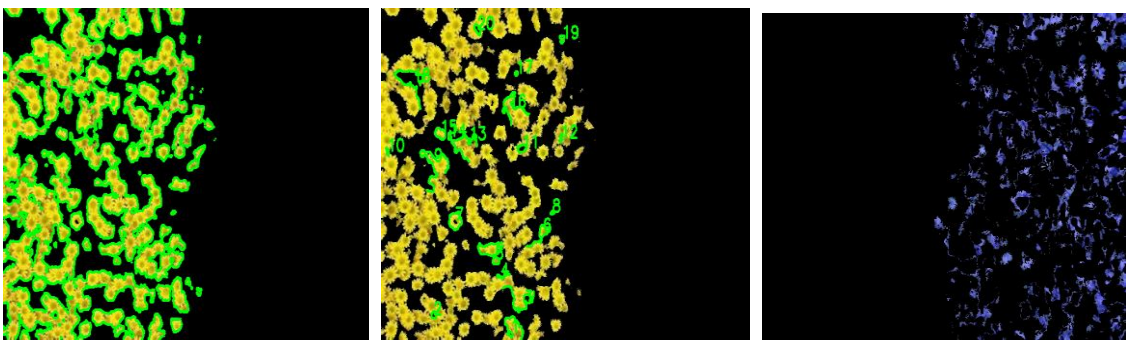




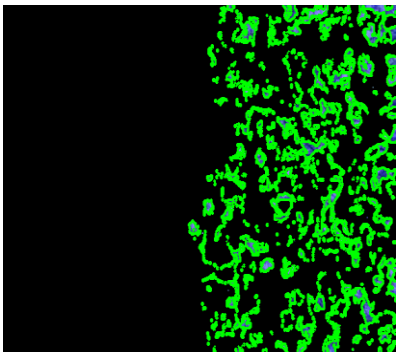
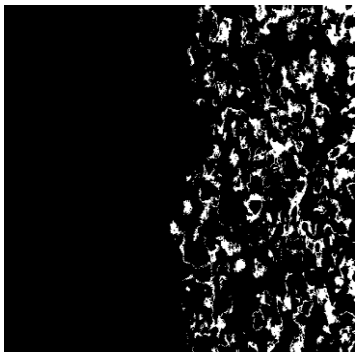
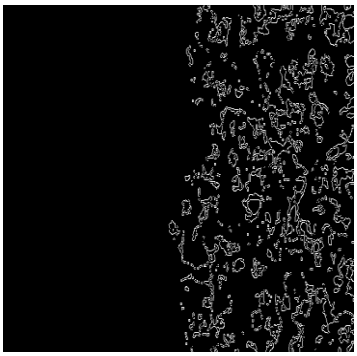
Result of Working Process:

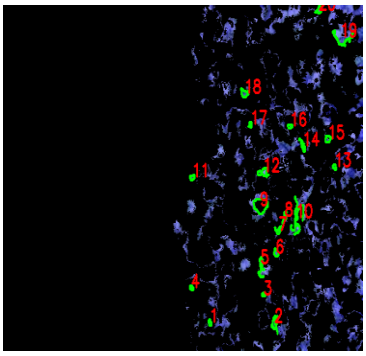
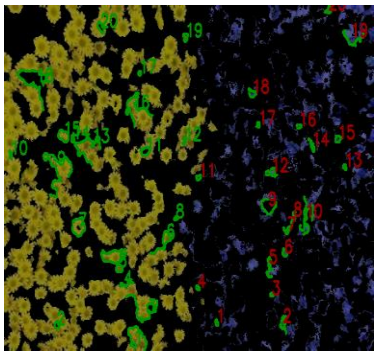






Result of Working Process:

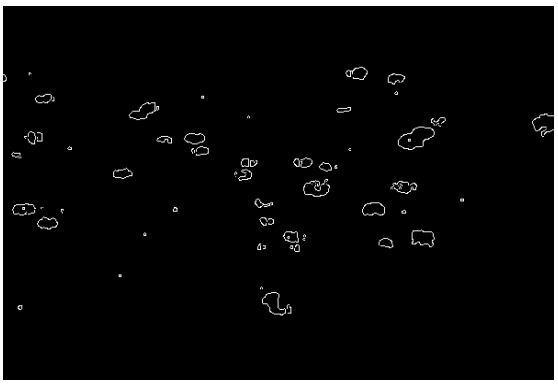
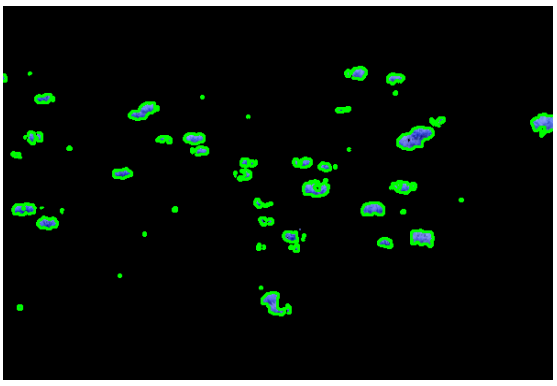
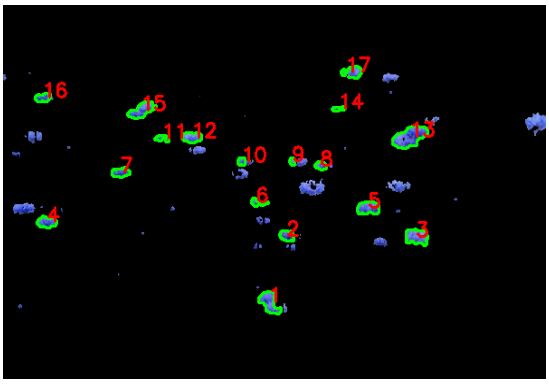



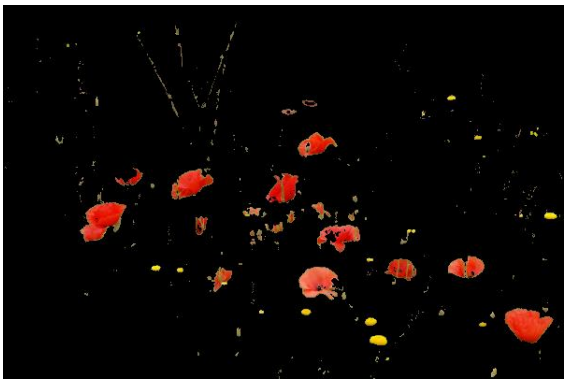
```
Main Colors and their Names:
Color: [30 43 31], Name: black
Color: [215 201  44], Name: yellow
Color: [119 116 205], Name: blue
<Figure size 1080x720 with 0 Axes>
Number of Yellow Flowers: 20
<Figure size 1080x720 with 0 Axes>
Number of Blue Flowers: 20
```

Result of Working Process:

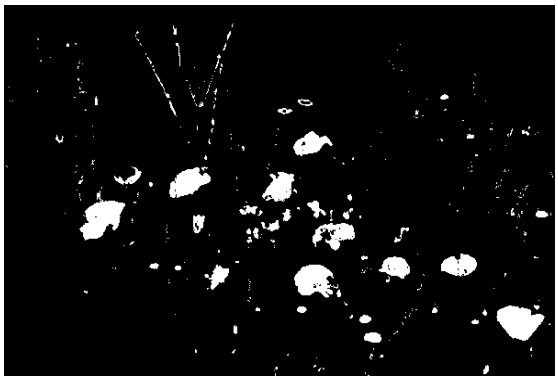


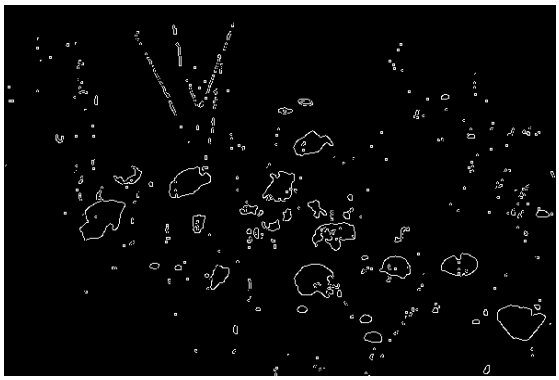


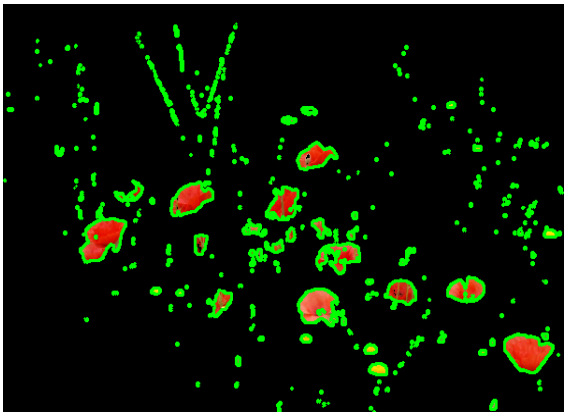




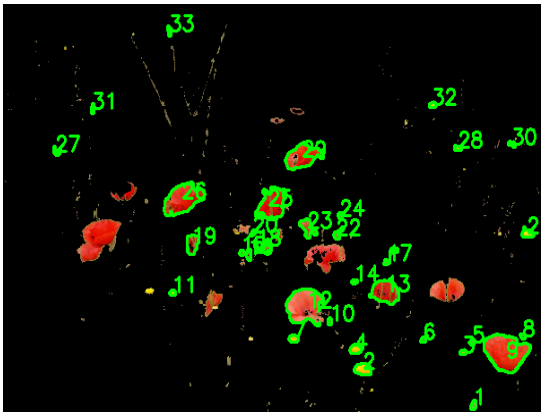


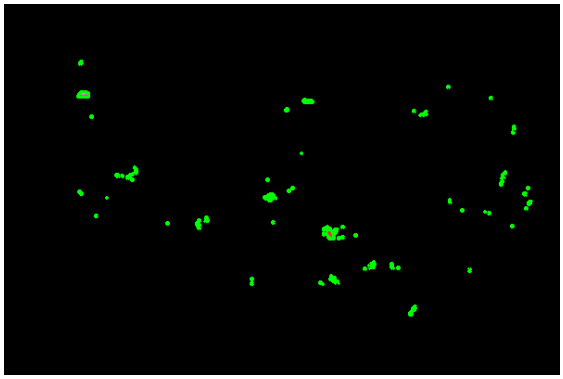
Result of Working Process:

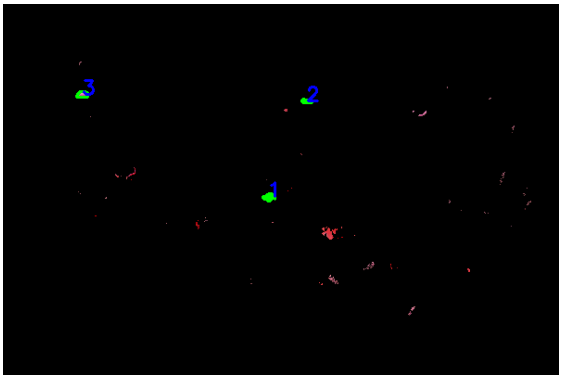






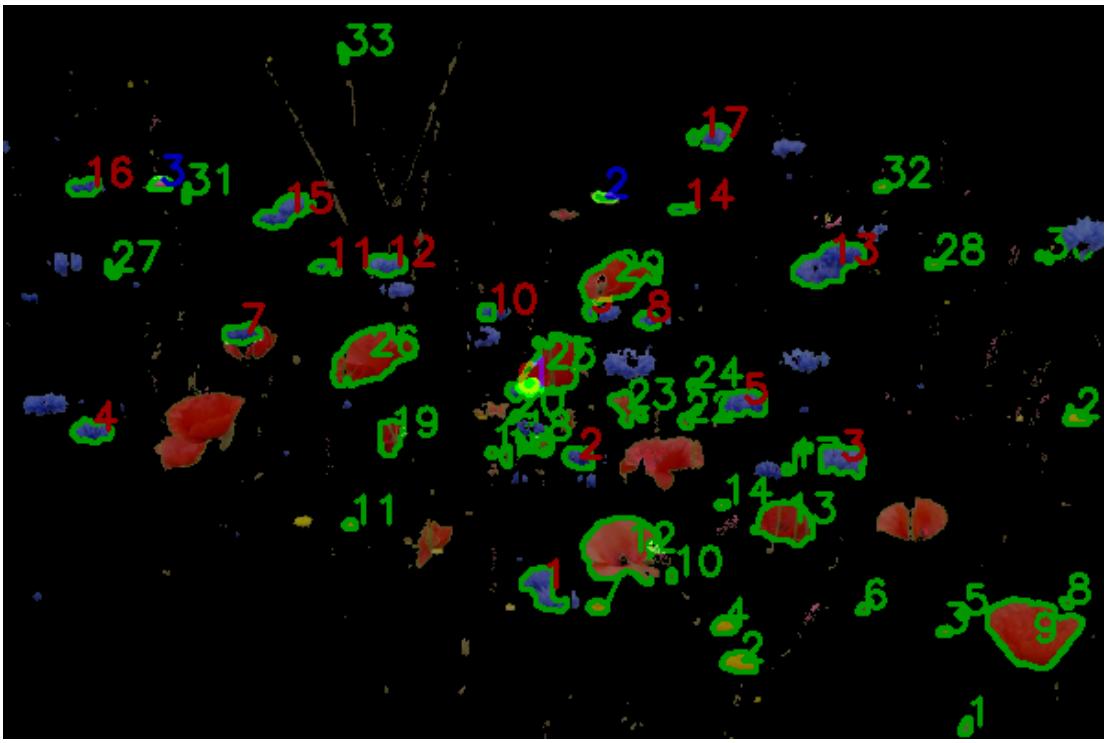







```
<Figure size 1080x720 with 0 Axes>
Number of Red Flowers: 3
<Figure size 1080x720 with 0 Axes>
Number of Blue Flowers: 17
<Figure size 1080x720 with 0 Axes>
Number of Yellow Flowers: 33
```

Result of Working Process:



Limitations:

- In case small or very small flowers, it hardly counts perfectly
- Sometimes, overlapping of flowers counted as one flower

Conclusion:

In conclusion, the combined use of K-shift clustering and color range masking significantly enhances the segmentation of flower colors and the extraction of main color themes from images. K-shift clustering dynamically adjusts centroids based on local data density, effectively capturing the complex color distributions and variations inherent in flower images. Color range masking further refines the process by selectively isolating specific hues relevant to the flowers, ensuring more targeted and accurate segmentation. This synergy improves the accuracy of color segmentation and facilitates the extraction of distinct and representative color palettes, making it invaluable for applications in botany, digital image processing, and visual content analysis across diverse datasets and real-world scenarios.