Image Blending and Subtraction

Image Blending:

Image blending combines two images using a weighted sum: I_3(r,c) = a * I_1(r,c) + (1-a) * I_2(r,c), where a is a blending factor (0 to 1). Applications include image compositing, transparency effects, image fusion (e.g., in medical imaging), and interpolation between images. OpenCV's cv2.addWeighted() implements this operation efficiently.

Image Subtraction for Change Detection:

Subtraction computes the absolute difference: I_3(r,c) = |I_1(r,c) - I_2(r,c)|. It highlights differences between images, useful for motion detection, change detection in surveillance, or medical imaging (e.g., identifying tumors). Non-zero pixel values in the result indicate changes.

-->Blending allows seamless integration of multiple images, enhancing visual effects or combining information.

-->Subtraction is critical for identifying temporal or spatial changes, enabling automated monitoring systems.

```
import cv2
import numpy as np
import matplotlib.pyplot as plt
```

Thresholding

```
# Load images
img1 = cv2.imread('/content/drive/MyDrive/Colab Notebooks/imageprocessingimages/everest.jpg'
img2 = cv2.imread('/content/drive/MyDrive/Colab Notebooks/imageprocessingimages/flag.jpg', 0
img3 = cv2.imread('/content/drive/MyDrive/Colab Notebooks/imageprocessingimages/rose.jpg', 1
img4 = cv2.imread('/content/drive/MyDrive/Colab Notebooks/imageprocessingimages/red.jpg', 1)
```

Masking Example

```
# Resize images
img1 = cv2.resize(img1, (250, 250))
img2 = cv2.resize(img2, (250, 250))
img3 = cv2.resize(img3, (250, 250))
img4 = cv2.resize(img4, (250, 250))
```

```
# Image blending using cv2.addWeighted
alpha = 0.3
beta = 1 - alpha
img_blended = cv2.addWeighted(img1, alpha, img2, beta, 0)
```

```
# Manual blending using the equation I_3(r,c) = a*I_1(r,c) + (1-a)*I_2(r,c)
image_addition = np.empty((250, 250))
for i in range(250):
    for j in range(250):
        image_addition[i][j] = alpha * img1[i][j] + beta * img2[i][j]
image_addition = np.round(image_addition).astype(np.uint8)
```

```
# Subtraction for change detection
img_subtraction = cv2.absdiff(img3, img4)
```

```
# Display results
plt.figure(figsize=(15, 10))

plt.subplot(2, 3, 1)
plt.imshow(img1, cmap='gray')
plt.title('Image 1 (Everest)')
plt.axis('off')

plt.subplot(2, 3, 2)
plt.imshow(img2, cmap='gray')
plt.title('Image 2 (Flag)')
plt.axis('off')

plt.subplot(2, 3, 3)
plt.imshow(img_blended, cmap='gray')
plt.title('Blended (cv2.addWeighted)')
plt.axis('off')

plt.subplot(2, 3, 4)
plt.imshow(image_addition, cmap='gray')
plt.title('Blended (Manual)')
plt.axis('off')

plt.subplot(2, 3, 5)
plt.imshow(img3[:, :, ::-1])  # Convert BGR to RGB
plt.title('Image 3 (Rose)')
plt.axis('off')

plt.subplot(2, 3, 6)
plt.imshow(img_subtraction[:, :, ::-1])  # Convert BGR to RGB
```

```
plt.title('Subtraction (Rose vs Modified)')
plt.axis('off')

plt.show()
```