CV Lecture Assignments - "Segmentation"

Overview

This document covers two image segmentation techniques used in computer vision: **Otsu's Thresholding** and **Adaptive Thresholding**. These methods are essential for separating objects from the background of grayscale images based on different thresholding strategies. The purpose of this task is to apply these techniques to process images and compare their results.

Prerequisites

- Python 3.x
- OpenCV (`cv2`)
- NumPy
- Matplotlib

Functions Description

1. Otsu's Thresholding

Otsu's Thresholding is a global thresholding technique that computes an optimal threshold by minimizing the intra-class variance of pixel values. It automatically determines the threshold value for image binarization.

```
def otsu_thresholding(image):
    # Apply Otsu's Thresholding method
    _, otsu_image = cv2.threshold(image, 0, 255, cv2.THRESH_BINARY +
cv2.THRESH_OTSU)
    return otsu_image
```

2. Adaptive Thresholding

Adaptive Thresholding applies local thresholding, where the threshold value is calculated for each pixel based on its local neighborhood. This method is useful for images with varying lighting conditions.

```
def adaptive_thresholding(image):
    # Apply adaptive thresholding using Gaussian mean
    adaptive_image = cv2.adaptiveThreshold(image, 255,
cv2.ADAPTIVE_THRESH_GAUSSIAN_C,
                                           cv2.THRESH_BINARY, 11, 2)
    return adaptive_image
```

Steps to Execute

1. Place the input grayscale image (`Input Grayscale Photography Of Woman.jpg`) in the same directory as the script.

2. Run the script to apply Otsu's Thresholding and Adaptive Thresholding to the image.
3. The output will display the original image along with the results of both thresholding techniques side by side.

Example Visualization

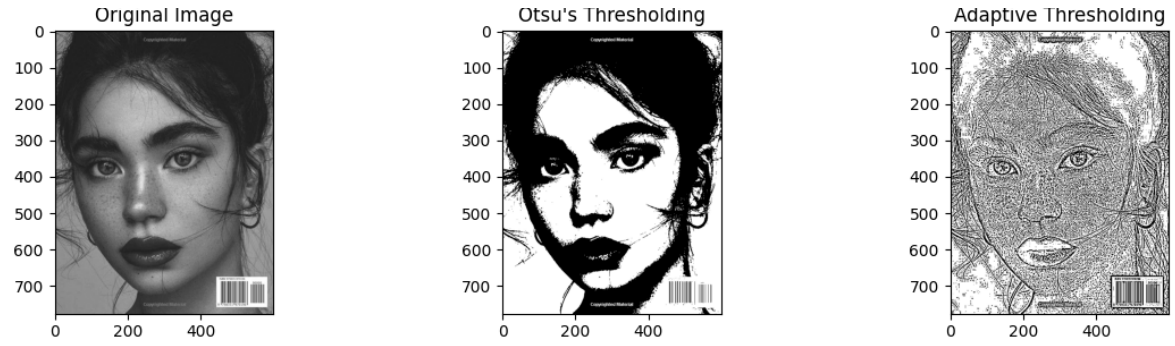
Displayed Images

- **Original Image**
- **Otsu's Thresholding**
- **Adaptive Thresholding**

Sample Visualization Code

```
plt.figure(figsize=(12, 10))

# Original Image
plt.subplot(2, 3, 1), plt.imshow(image, cmap='gray'), plt.title('Original Image')

# Otsu's Thresholding
plt.subplot(2, 3, 2), plt.imshow(otsu_image, cmap='gray'), plt.title("Otsu's
Thresholding")

# Adaptive Thresholding
plt.subplot(2, 3, 3), plt.imshow(adaptive_image, cmap='gray'), plt.title('Adaptive
Thresholding')

plt.tight_layout()
plt.show()
```

Output Example



Comparison of Techniques

1. Otsu's Thresholding

- **Purpose**: Finds the optimal threshold for global binarization of the image.
- **Strengths**: Automatically determines the threshold, making it ideal for images with clear contrast between foreground and background.
- **Weaknesses**: May not work well on images with uneven lighting or multiple foreground objects.

2. Adaptive Thresholding

- **Purpose**: Applies local thresholding to segments of the image, adjusting the threshold dynamically for each pixel.
- **Strengths**: Effective for images with varying lighting conditions or complex backgrounds.
- **Weaknesses**: Requires more computational resources and may introduce noise in images with uniform regions.

Summary

Both Otsu's and Adaptive Thresholding are powerful techniques for image segmentation. Otsu's method works well for globally thresholding images with distinct foreground and background contrasts, while Adaptive Thresholding provides better results for images with uneven lighting or intricate details.

By Ahmed Ibrahim Metwally Negm - ID: 322223887

Supervisors: Dr. Shimaa Othman

Thank you, ✨ 💟