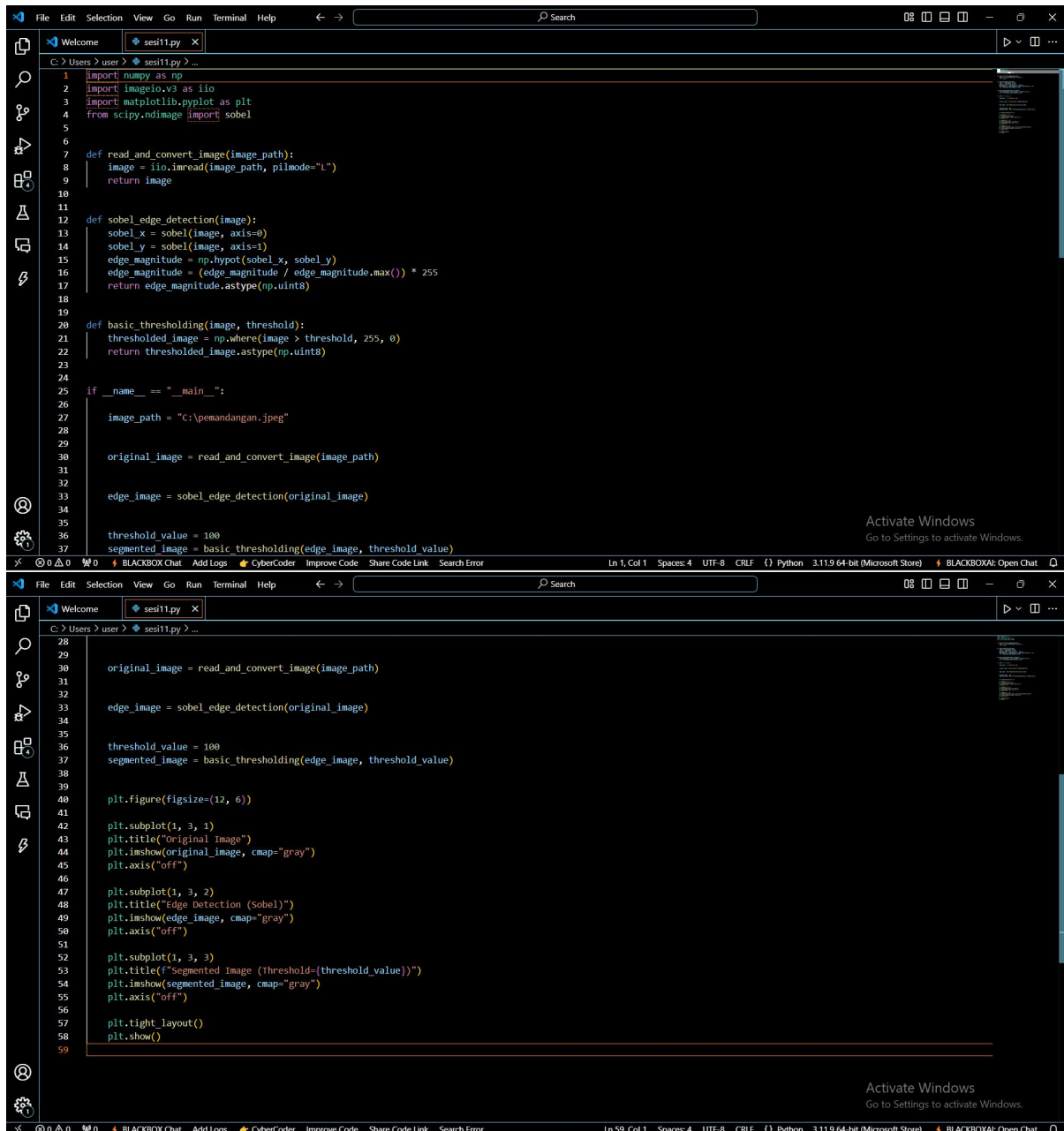


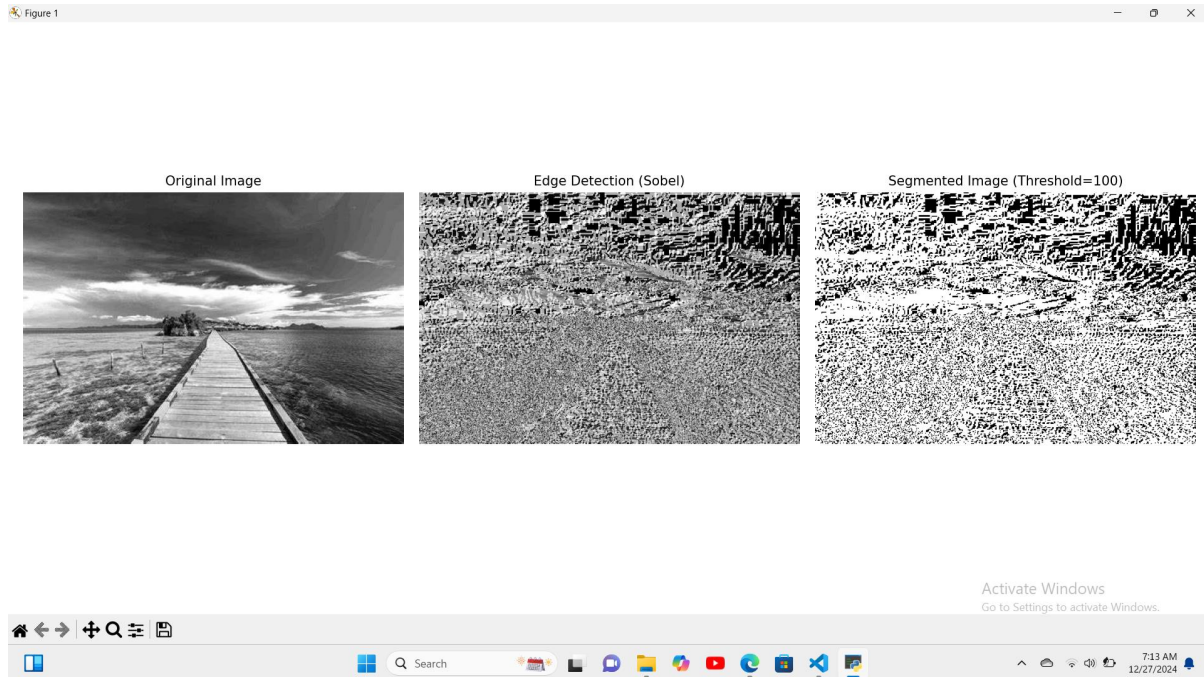
Nama : Hilda Deliana A
Nim : 20220040112
Kelas : TI22J

Tugas Sesi 13

Buatlah implementasi program segmentasi citra dengan memanfaatkan imageio, numpy dan matplotlib untuk basic thresholding dengan input gambar yang sudah dilakukan deteksi tepi menggunakan sobel. Lakukan analisa dari hasil output yang kalian dapatkan



```
1 import numpy as np
2 import imageio.v3 as iio
3 import matplotlib.pyplot as plt
4 from scipy.ndimage import sobel
5
6
7 def read_and_convert_image(image_path):
8     image = iio.imread(image_path, pilmode="L")
9     return image
10
11
12 def sobel_edge_detection(image):
13     sobel_x = sobel(image, axis=0)
14     sobel_y = sobel(image, axis=1)
15     edge_magnitude = np.hypot(sobel_x, sobel_y)
16     edge_magnitude = (edge_magnitude / edge_magnitude.max()) * 255
17     return edge_magnitude.astype(np.uint8)
18
19
20 def basic_thresholding(image, threshold):
21     thresholded_image = np.where(image > threshold, 255, 0)
22     return thresholded_image.astype(np.uint8)
23
24
25 if __name__ == "__main__":
26
27     image_path = "C:\\pemandangan.jpeg"
28
29     original_image = read_and_convert_image(image_path)
30
31
32     edge_image = sobel_edge_detection(original_image)
33
34
35     threshold_value = 100
36     segmented_image = basic_thresholding(edge_image, threshold_value)
37
38
39
40
41 plt.figure(figsize=(12, 6))
42
43 plt.subplot(1, 3, 1)
44 plt.title("Original Image")
45 plt.imshow(original_image, cmap="gray")
46 plt.axis("off")
47
48 plt.subplot(1, 3, 2)
49 plt.title("Edge Detection (Sobel)")
50 plt.imshow(edge_image, cmap="gray")
51 plt.axis("off")
52
53 plt.subplot(1, 3, 3)
54 plt.title("Segmented Image (Threshold=(threshold_value))")
55 plt.imshow(segmented_image, cmap="gray")
56 plt.axis("off")
57
58 plt.tight_layout()
59 plt.show()
```



Analisis Hasil Segmentasi Citra

Citra Asli (Grayscale)

Gambar asli dikonversi ke skala abu-abu untuk menyederhanakan proses analisis dan deteksi tepi. Grayscale membantu fokus pada intensitas piksel tanpa memengaruhi warna, yang relevan untuk operator Sobel. Konversi ini juga mempercepat pemrosesan dengan mengurangi kompleksitas data.

Hasil Deteksi Tepi dengan Sobel

Operator Sobel mendeteksi perubahan intensitas pada arah horizontal dan vertikal, menghasilkan peta gradien yang menunjukkan kekuatan perubahan intensitas. Bagian tepi objek yang jelas menjadi lebih terlihat, sedangkan area homogen tetap gelap.

Keuntungan: Sobel bekerja dengan cepat dan memberikan hasil deteksi tepi yang tajam.

Kelemahan: Sobel cenderung terpengaruh oleh noise, sehingga diperlukan pengolahan awal seperti penghalusan untuk hasil yang lebih bersih.

Hasil Segmentasi dengan Thresholding

Thresholding sederhana digunakan untuk memisahkan tepi yang kuat (foreground) dari latar belakang. Nilai ambang menentukan seberapa sensitif proses segmentasi terhadap intensitas gradien.

Pemilihan Threshold: Threshold tinggi hanya menampilkan tepi yang paling signifikan, sementara threshold rendah mempertahankan lebih banyak detail namun bisa menambahkan noise.

Keuntungan: Teknik ini cepat dan efektif untuk gambar dengan kontras tinggi.

Kelemahan: Tidak adaptif terhadap perubahan intensitas lokal, sehingga hasilnya kurang optimal pada gambar dengan pencahayaan yang tidak merata.

Analisis Visual

Citra Tepi: Menampilkan garis terang di sekitar area objek, yang menjadi acuan untuk segmentasi.

Citra Tersegmentasi: Menghasilkan gambar biner, di mana foreground mewakili area objek yang dideteksi berdasarkan threshold yang diterapkan.

Evaluasi Hasil Segmentasi

Keberhasilan: Teknik ini efektif untuk gambar dengan kontras yang jelas antara objek dan latar belakang.

Keterbatasan: Pada gambar dengan kompleksitas tinggi atau noise yang signifikan, hasil segmentasi bisa kurang presisi, memerlukan metode tambahan seperti thresholding adaptif atau pre-processing.

Link github : https://github.com/HildaDeliana/sesi_11..git