
TUTORIAL 5

Image Segmentation

Pengolahan Citra - Semester Gasal 2023/2024

1 Introduction

Image segmentation atau segmentasi citra merupakan teknik yang sering digunakan pada pemrosesan citra digital. Tujuannya adalah untuk mempartisi suatu citra ke dalam beberapa bagian atau wilayah (biasanya berdasarkan karakteristik piksel pada citranya) [1].

Menurut [2], terdapat dua pendekatan pada segmentasi citra, yakni:

1. **Discontinuity Approach:** Memanfaatkan diskontinuitas (perbedaan mencolok) dari intensitas piksel. Contoh metode segmentasi dengan pendekatan ini adalah *edge-based segmentation*.
2. **Similarity Approach:** Memanfaatkan similaritas antarpiksel. Contoh metode segmentasi dengan pendekatan ini adalah *distributed-based segmentation* dan *region-based segmentation*.

2 Edge-Based Segmentation

2.1 Basic Edge Detection

Pada metode ini, kita menggunakan algoritma-algoritma *edge detection* dasar seperti *Roberts*, *Prewitt*, dan *Sobel* [2]. Anda dapat mengombinasikan hasilnya dengan *thresholding* untuk mendapatkan *edge* yang "signifikan" (intensitasnya tinggi) [2]. Berikut adalah contoh kode dan hasilnya. Sebagai catatan, `threshold cv2.THRESH_BINARY` membuat nilai piksel pada rentang yang diberikan sama dengan intensitas terbesar pada citra. Anda dapat mencoba rentang berbeda.

```
1 # Import packages
2 import cv2
3 import numpy as np
4 from matplotlib import pyplot as plt
5 from skimage import io, color, filters, util
6 from skimage.morphology import disk, ball
7
8 img = io.imread('/content/valve.png')
9 gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
10 sobel = util.img_as_ubyte(filters.sobel(gray))
11
12 # Anda dapat bereksperimen dengan nilai lower and upper bound
13 # lain dari threshold
14 _, sobel_threshold = cv2.threshold(sobel, 15, 255,
15 cv2.THRESH_BINARY)
16
17 plt.figure(figsize=(15,7.5))
18 plt.subplot(131),plt.imshow(img)
19 plt.title('Original Image')
20 plt.subplot(132),plt.imshow(sobel, cmap='gray')
21 plt.title('Sobel Image')
```

```

20 plt.subplot(133),plt.imshow(sobel_threshold, cmap='gray')
21 plt.title('Sobel Thresholded Image')
22 plt.show()

```

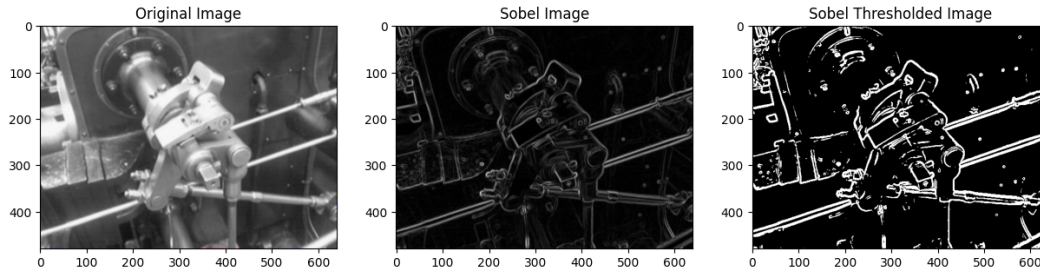


Figure 1: Contoh *Basic Edge Detection* ([Sumber Citra](#))

2.2 Canny Edge Detection

Pada metode ini, dilakukan deteksi *edges* sekaligus menekan *noise* yang ada pada saat bersamaan [2]. Berikut adalah contoh kode dan hasilnya. Sebagai catatan, Anda harus menentukan *low* dan *high threshold* pada fungsi `cv2.Canny()`. Menurut [tutorial ini](#), disarankan rasio antara keduanya adalah 1:2 atau 1:3.

```

1  # Import packages
2  import cv2
3  import numpy as np
4  from matplotlib import pyplot as plt
5  from skimage import io, color, filters, util
6  from skimage.morphology import disk, ball
7
8  img = io.imread('/content/valve.png')
9  gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
10
11 # Anda dapat bereksperimen dengan nilai low dan high threshold
   lainnya, disarankan
12 # rasio antara low dan high threshold 1:2 atau 1:3.
13 canny = cv2.Canny(gray, 55, 165)
14
15 plt.figure(figsize=(15,7.5))
16 plt.subplot(121),plt.imshow(img)
17 plt.title('Original Image')
18 plt.subplot(122),plt.imshow(canny, cmap='gray')
19 plt.title('Canny Image')
20 plt.show()

```

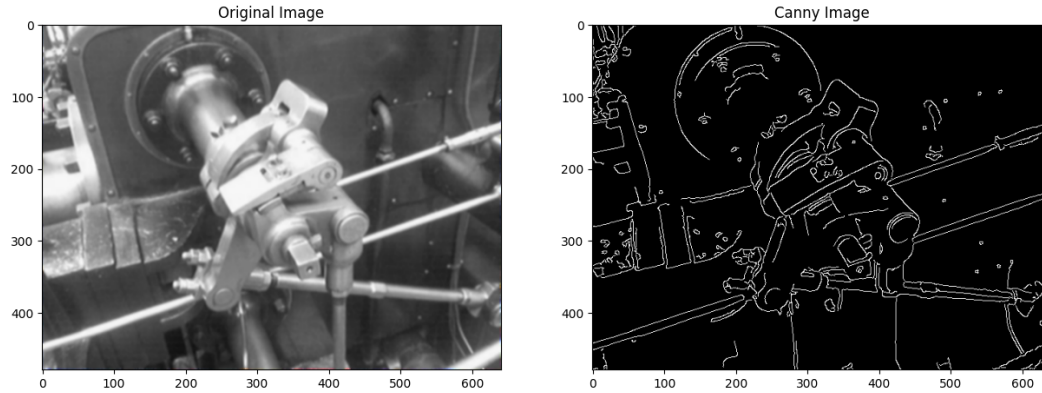


Figure 2: Contoh *Canny Edge Detection* ([Sumber Citra](#))

3 *Distribution-Based Segmentation*

3.1 *Otsu's Thresholding*

Metode ini merupakan salah satu contoh *distribution-based segmentation*. Pada metode ini, idenya adalah menentukan secara otomatis nilai *threshold* yang tepat, kemudian tinggal melakukan *thresholding* pada citra dengan nilai tersebut [2]. Berikut adalah contoh kode dan hasilnya.

```

1  # Import packages
2  import cv2
3  import numpy as np
4  from matplotlib import pyplot as plt
5  from skimage import io, color, filters, util
6  from skimage.morphology import disk, ball
7
8  img_thresh = io.imread('ingredient.jpg')
9  G = util.img_as_ubyte(color.rgb2gray(img_thresh))
10 T = filters.threshold_otsu(G)
11 S = util.img_as_float(G > T)
12
13 plt.figure(figsize=(20,10))
14 plt.subplot(1,2,1); plt.imshow(img_thresh)
15 plt.title('Original'); plt.axis("off")
16 plt.subplot(1,2,2); plt.imshow(1-S, cmap='gray')
17 plt.title("Thresholded"); plt.axis("off")
18 plt.show()

```



Figure 3: Contoh *Otsu's Thresholding* ([Sumber Citra](#))

4 *Region-Based Segmentation*

4.1 *Simple Linear Iterative Clustering (SLIC)*

Metode ini merupakan salah satu contoh *region-based segmentation*. Pada metode ini, kita membagi citra ke kumpulan superpiksel yang homogen. Sederhananya, metode ini menyerupai algoritma *K-Nearest Neighbor* (KNN). Berikut adalah contoh kode beserta hasilnya. Sebagai catatan, jumlah segmenn yang dihasilkan tidak mesti sebanyak `n_segments` karena dibatasi juga jumlah iterasinya. Silakan merujuk ke dokumentasi `skimage` untuk mengetahui parameter apa saja yang ada.

```

1  # Import packages
2  import cv2
3  import numpy as np
4  from matplotlib import pyplot as plt
5  from skimage import io, color, filters, util
6  from skimage.morphology import disk, ball
7  from skimage.color import rgb2gray
8  from skimage.filters import sobel
9  from skimage.segmentation import mark_boundaries, slic
10 from skimage.util import img_as_float
11
12 img2 = io.imread('cat.jpg')
13
14 segments_slic = slic(img2, n_segments=200)
15
16 print(f"SLIC number of segments:
17       {len(np.unique(segments_slic))}")
18
19 plt.figure(figsize=(10,7))
20 plt.imshow(mark_boundaries(img2, segments_slic))
21 plt.title('SLIC')
22 plt.show()

```

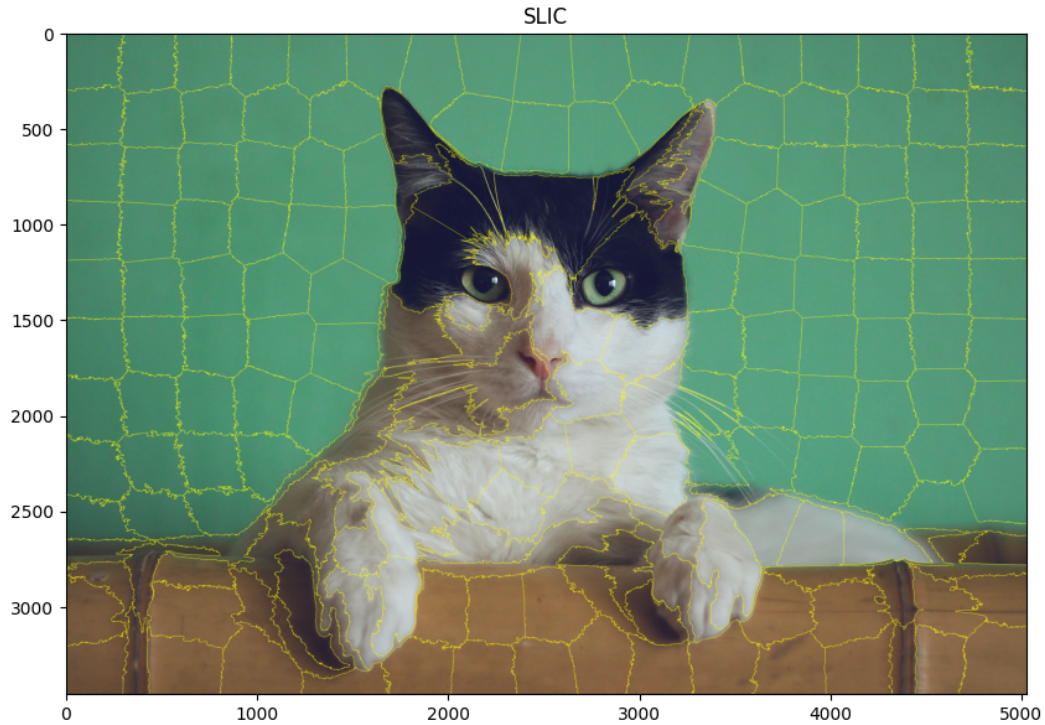


Figure 4: Contoh Segmentasi SLIC ([Sumber Citra](#))

Lebih lanjut, kita bisa menyeragamkan nilai-nilai piksel pada tiap superpiksel dengan nilai tertentu (misal dengan rata-ratanya). Berikut adalah contoh kode dan hasilnya.

```

1  def superpixel_mean_image(image, slic_labels):
2      """
3          Code taken from https://stackoverflow.com/a/57746835
4          Turn image into superpixel image, given SLIC segment
5              labels.
6      """
7      im_rp = image.reshape(
8          (image.shape[0]*image.shape[1],image.shape[2]))
9      sli_1d=np.reshape(slic_labels,-1)
10     uni=np.unique(sli_1d)
11     new_img=np.zeros(im_rp.shape)
12     for i in uni:
13         loc=np.where(sli_1d==i)[0]
14         #print(loc)
15         mm=np.mean(im_rp[loc,:],axis=0)
16         new_img[loc,:]=mm
17     return np.reshape(
18         new_img,[image.shape[0],
19             image.shape[1],
20             image.shape[2]]).astype('uint8')

```

```

20
21 superpixel_image = superpixel_mean_image(img2, segments_slic)
22
23 plt.figure(figsize=(10,7))
24 plt.imshow(superpixel_image)
25 plt.show()

```



Figure 5: Contoh Penyeragaman Pikel SLIC ([Sumber Citra](#))

References

- [1] M. Team, "What is image segmentation? 3 things you need to know," 2023. [Online]. Available: <https://www.mathworks.com/discovery/image-segmentation.html>
- [2] A. "Arymurthy, L. Rahadiani, and M. Rachmadi, "Image segmentation," pPT Pengcit.