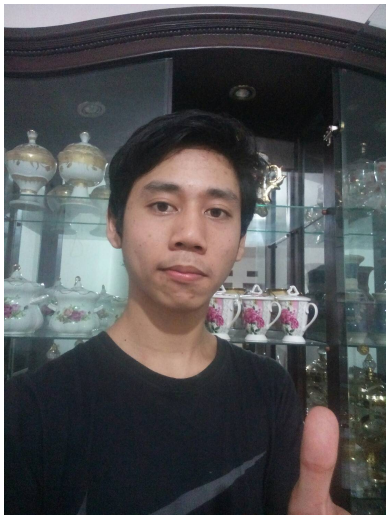
LAPORAN
TUGAS BESAR 3
IF4073
Interpretasi dan Pengolahan Citra




13519068 Roy H Simbolon

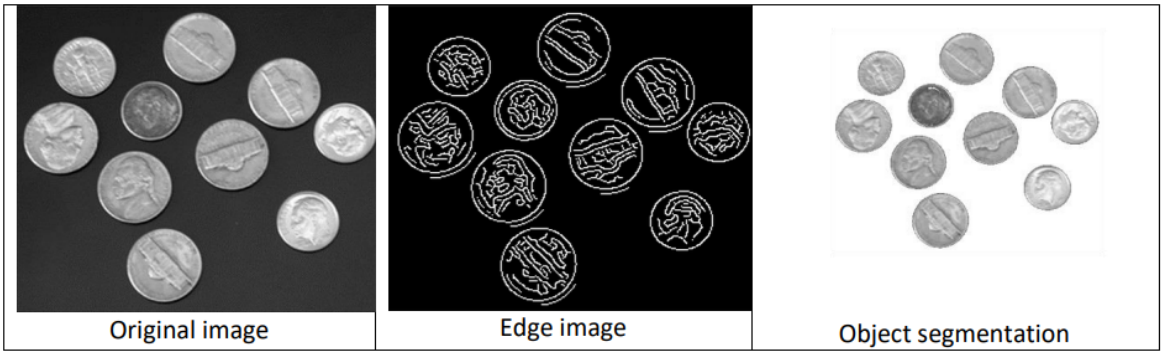
13519203 R. B. Wishnumurti

PRODI STUDI TEKNIK INFORMATIKA
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG

2022

1. Spesifikasi Tugas

Salah satu kegunaan deteksi tepi adalah untuk segmentasi objek, yaitu mendeteksi objek melalui bentuknya. Bentuk objek dapat diperoleh dari hasil pendeteksian tepi. Setelah tepi objek dideteksi, selanjutnya objek dipisahkan dari latar belakangnya, untuk kemudian digunakan dalam proses pengenalan objek (object recognition).

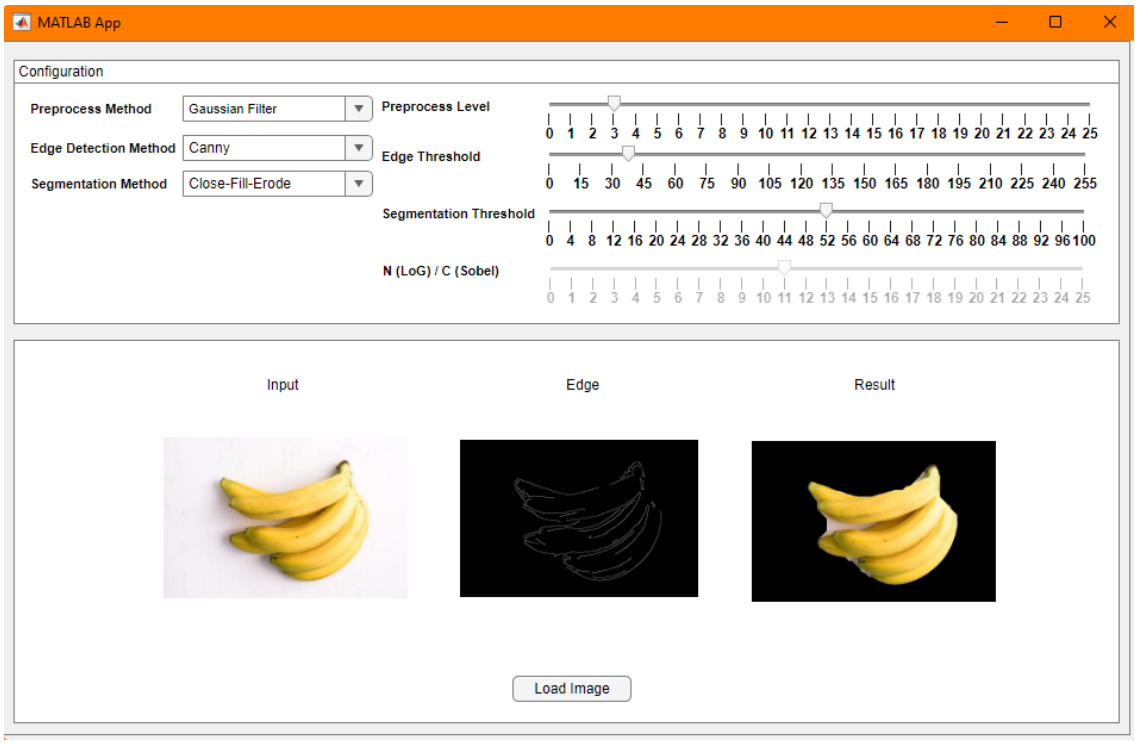


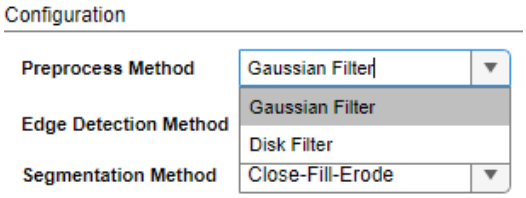
Pada Tugas 3 ini, penulis diminta membuat program Matlab untuk mensegmentasi objek buah-buahan dari latar belakangnya berdasarkan hasil pendeteksian tepi. Pendeteksian tepi menggunakan bermacam macam operator deteksi tepi: Laplace, LoG, Sobel, Prewitt, Roberts, Canny, namun tidak boleh menggunakan fungsi built-in deteksi tepi yang terdapat di dalam Matlab (kecuali operator Canny).

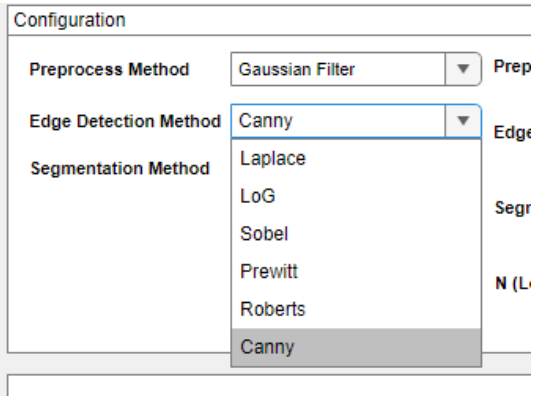
Menurut penulis, adapun tugas ini dibuat untuk mencapai tujuan utama berikut:

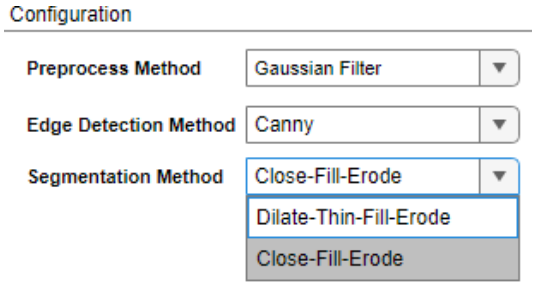
1. Menghasilkan gambar biner tepi dari inputan gambar dengan menggunakan deteksi tepi Laplace, Laplacian of Gaussian, Sobel, Prewitt, Roberts, dan Canny.
2. Menggunakan gambar biner tepi untuk mensegmentasi objek yang ada pada gambar dari latar belakang.

2. GUI Program









3. Program

3.1. Kode Program

Fungsi Laplace

```
function Out = Laplace(I, threshold)
    H = [0 1 0; 1 -4 1; 0 1 0];
    Out = uint8(convn(double(I), double(H), "same")) > threshold;
end
```

Fungsi Laplacian

```
function Out = LoG(I, n0, threshold)
    n = round(n0);
    s = n/5;
    r = (n-1) / 2;
    X = repmat(-r:r, n, 1);
    Y = repmat(transpose(-r:r), 1, n);
    t = -(X.*X+Y.*Y)/(2.0*s*s);
    H = (-1/(pi*s*s*s*s)) * ((1+t) .* exp(t));
    H = H - mean2(H);
    %H = fspecial('log', n, s)
    Out = uint8(convn(double(I), double(H), "same")) > threshold;
end
```

Fungsi Sobel

```
function Out = Sobel(I, c0, threshold)
    c = round(c0);
    Hx = [-1 0 1; -c 0 c; -1 0 1];
    Hy = [1 c 1; 0 0 0; -1 -c -1];
    Jx = convn(double(I), double(Hx), "same");
    Jy = convn(double(I), double(Hy), "same");
    Out = uint8(abs(Jx) + abs(Jy)) > threshold;
end
```

Fungsi Prewitt

```
function Out = Prewitt(I, threshold)

    Hx = [-1 0 1; -1 0 1; -1 0 1];
    % Filter vertikal
    Hy = [1 1 1; 0 0 0; -1 -1 -1];
    Jx = convn(double(I), double(Hx), "same");
    Jy = convn(double(I), double(Hy), "same");
    % Jumlahkan kedua hasil konvolusi
    Out = uint8(abs(Jx) + abs(Jy) > threshold);
end
```

Fungsi Roberts

```
  function Out = Roberts(I, threshold)
      Hx = [1 0; 0 -1];
      Hy = [0 1; -1 0];
      Jx = convn(double(I), double(Hx), "same");
      Jy = convn(double(I), double(Hy), "same");
      Out = uint8(abs(Jx) + abs(Jy)) > threshold;
  end
```

Fungsi Canny

```
function Out = Canny(I, threshold)
    Out = edge(I, 'canny', threshold/255);
end
```

Fungsi Bluring

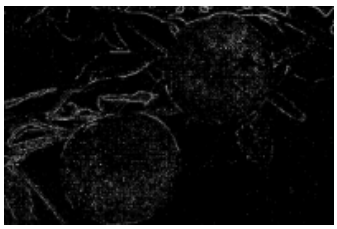
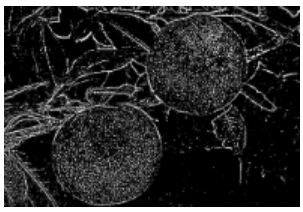
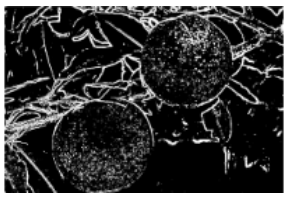
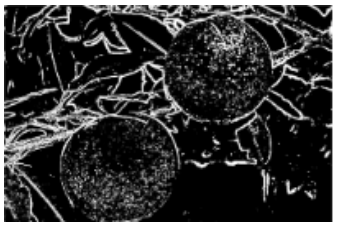
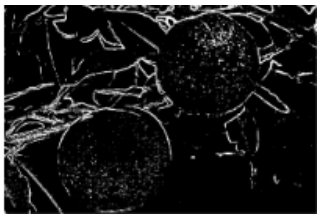
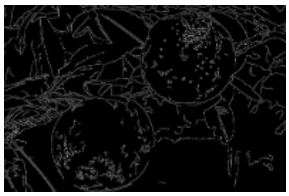
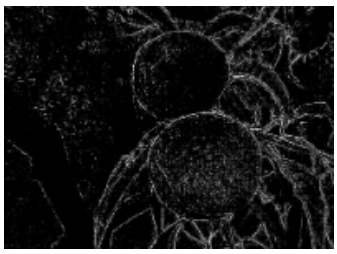
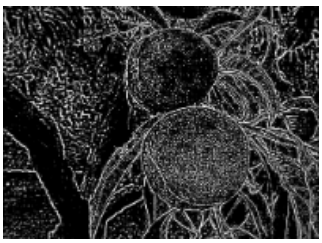
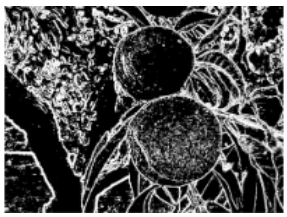
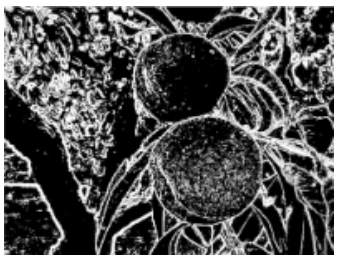
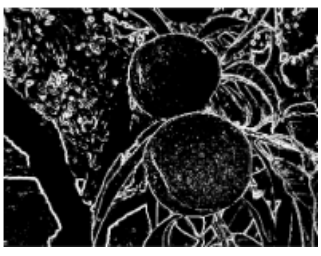
```
% Fungsi blurring (pemrosesan awal)
function result = GenerateBlurImage(~, inputImage, edgeDetectionMethod, aLevel)
    if (aLevel > 0)
        switch edgeDetectionMethod

            % Blurring dengan filter disk
            case 'Disk Filter'
                result = imfilter(inputImage, fspecial('disk', aLevel), 'replicate');

            % Blurring dengan filter gaussian
            case 'Gaussian Filter'
                result = imgaussfilt(inputImage, aLevel);

            otherwise
                result = inputImage;
        end
    else
        result = inputImage;
    end
end
```

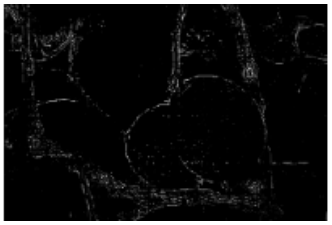
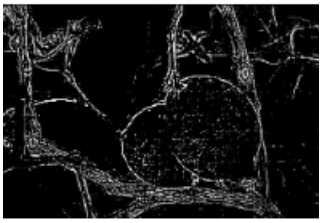
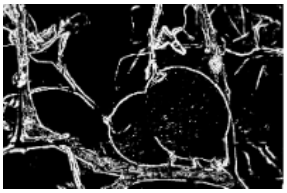
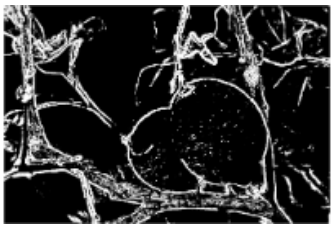
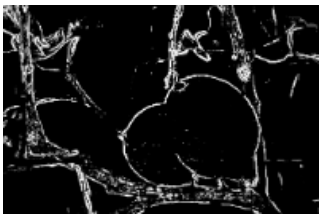
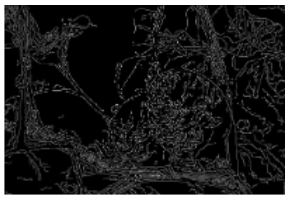
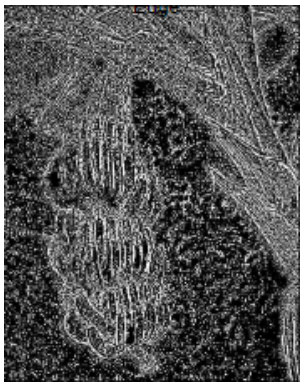
Fungsi Biner Mask

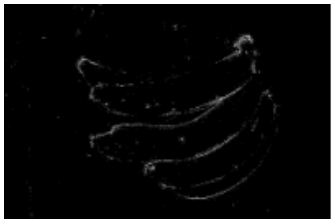
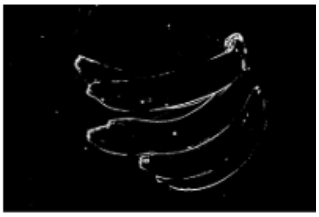
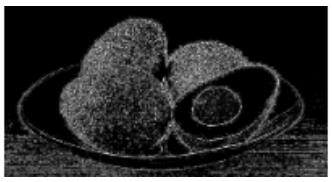
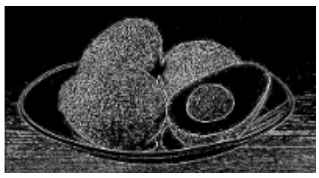
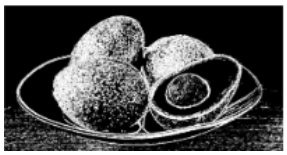
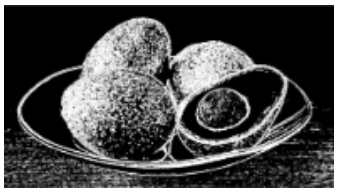
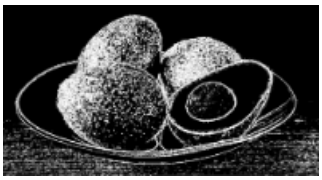
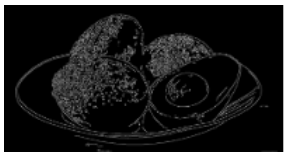
```
% Fungsi pembuatan gambar biner mask
function result = GenerateMaskImage(~, inputImage, edgeDetectionMethod, threshold)
    switch edgeDetectionMethod

        % Buat mask dengan metode 1, untuk gambar dengan banyak
        % tepi
        case 'Dilate-Thin-Fill-Erode'


            cleanImage = imclearborder(inputImage);

            cleanImage = imdilate(cleanImage, strel('disk', round(threshold)));
            cleanImage = bwmorph(cleanImage, 'thin', Inf);


            cleanImage = imfill(cleanImage, 'holes');

            cleanImage = imerode(cleanImage, strel('disk', 2));

            result = uint8(cleanImage);

        % Buat mask dengan metode 2, untuk gambar sederhana
        case 'Close-Fill-Erode'


            cleanImage = imclearborder(inputImage);

            cleanImage = imclose(cleanImage, strel('disk', round(threshold)));

            cleanImage = imfill(cleanImage, 8, 'holes');

            cleanImage = imerode(cleanImage, strel('disk', 2));

            result = uint8(cleanImage);

        otherwise
            result = uint8(ones([size(inputImage, 1), size(inputImage, 2)]));
    end
end
```

3.2. Hasil Eksekusi

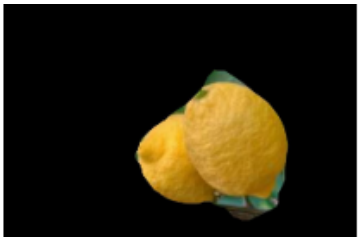
3.2.1 Deteksi Tepi

<table>
<tr><td>Inputan gambar</td><td colspan="3">Method deteksi tepi</td></tr>
<tr><td rowspan="4">  </td><td>Laplace (T=8)</td><td>Laplacian of Gaussian (N=5, T=2)</td><td>Sobel (T=67, C=3)</td></tr>
<tr><td>  </td><td>  </td><td>  </td></tr>
<tr><td>Prewitt (T=38)</td><td>Roberts (T=15)</td><td>Canny (T=30)</td></tr>
<tr><td>  </td><td>  </td><td>  </td></tr>
<tr><td rowspan="4">  </td><td>Laplace (T=8)</td><td>Laplacian of Gaussian (N=5, T=2)</td><td>Sobel (T=67, C=3)</td></tr>
<tr><td>  </td><td>  </td><td>  </td></tr>
<tr><td>Prewitt (T=38)</td><td>Roberts (T=15)</td><td>Canny (T=30)</td></tr>
<tr><td>  </td><td>  </td><td>  </td></tr>
</table>

	Laplace (T=8)	Laplacian of Gaussian (N=5, T=2)	Sobel (T=67, C=3)
			
	Prewitt (T=38)	Roberts (T=15)	Canny (T=30)
			
	Laplace (T=8)	Laplacian of Gaussian (N=5, T=2)	Sobel (T=67, C=3)
			
	Prewitt (T=38)	Roberts (T=28)	Canny (T=30)
			

	Laplace (T=8)	Laplacian of Gaussian (N=5, T=2)	Sobel (T=67, C=3)
			
	Prewitt (T=38)	Roberts (T=15)	Canny (T=30)
			
	Laplace (T=8)	Laplacian of Gaussian (N=5, T=2)	Sobel (T=67, C=3)
			
	Prewitt (T=38)	Roberts (T=15)	Canny (T=30)
			

3.2.2 Segmentasi

Inputan gambar	Hasil segmentasi	Parameter
		**Preprocess:** - **Deteksi Tepi :** Prewitt, T = 37 **Segmentasi:** Close-Fill-Erode, T = 10
		**Preprocess:** - **Deteksi Tepi :** Laplacian of Gaussian, N=3, T = 15 **Segmentasi:** Dilate-Thin-Fill-Erode, T = 13
		**Preprocess:** - **Deteksi Tepi :** Canny, T = 173 **Segmentasi:** Dilate-Thin-Fill-Erode, T = 64
		**Preprocess:** - **Deteksi Tepi :** Canny, T = 75 **Segmentasi:** Close-Fill-Erode, T = 39

		**Preprocess:** 3 **Deteksi Tepi :** Canny,T = 39 **Segmentasi:** Close-Fill-Erode, T = 52
		**Preprocess:** - **Deteksi Tepi :** Canny, T = 102 **Segmentasi:** Dilate-Thin-Fill-Erode, T = 15

4. Alamat Github

Kode lengkap untuk program tugas ini dapat diakses di link berikut:
https://github.com/bhanuharya/tugas-3-citra