Lab5-General

September 2, 2020

[1]:
```
pkg load image
```

[20]:
```
function padded = padding(matrix, type_of, mask_size)

    % Como usar esta funcion
    % matrix es el valor de la imagen (si es a color, debera ser convertida a␣
↪escala de grises o binarizarse)
    % type_of es el tipo de padding, 1 es el padding con reflejo, 2 es el␣
↪padding con zeros (recomendado
    % para morfologia)
    % mask_size  es el TAMANO de la mascara, no la mascara, por que si se desea␣
↪pasar la mascara, se debe usar
    % size(<mascara>) , nunca pasar la mascara.

    [x,y] = size(matrix);
    m_x = mask_size(1);
    m_y = mask_size(2);

    if m_x == m_y
        n_ref = (m_x - 1) / 2;

        % Mirrored
        if type_of == 1
            temp = matrix;

            % left - right
            temp = [fliplr(matrix(:,1:n_ref)) , matrix, fliplr(matrix)(:,1:
↪n_ref)];
            temp = [fliplr(rot90(temp(1:n_ref,:),2)) ; temp ;␣
↪fliplr(rot90(temp,2)(1:n_ref,:))];
            padded = temp;
        % Zero-ed
        elseif type_of == 2
            temp = zeros(x + 2 * n_ref, y + 2 * n_ref);
            temp(1 + n_ref:end - n_ref,1 + n_ref:end - n_ref) = matrix;
            padded = temp;
        else
```

```
            padded = zeros(x,y);
        end
    else
        padded = zeros(x,y);
    end
end
```

[21]:
```
function morph_matrix = morphologic_converter(image, operation,␣
↪structural_element, gray)

    % como usar esta funcion
    % image es la imagen CON PADDING, no usar esta funcion sin aplicar el␣
↪padding primero
    % operation es la operacion fundamental a aplicar : 1 es Dilatacion , 2 es␣
↪Erosion
    % structural_element es , como su nombre dice, el elemento estructural, NO␣
↪USAR TAMANOS PARES (3, 5, 7)
    % gray es un valor binario (true / false) , seleccionar true ejectua la␣
↪funcion en modo escala de grises
    % ejecutarla usando el valor false, trabaja de forma binaria.

    gen_img= image;

    [x,y]=size(gen_img);
    morph_matrix=zeros(x,y);

    n_ref = (size(structural_element)(1) - 1) / 2;
    struct_indx = find(structural_element);
    struct_sum = sum(sum(structural_element));

    if gray
        if operation == 1
            %dilatacion

            for s=1+n_ref:x-n_ref
                for t=1+n_ref:y-n_ref
                    temp = gen_img(s-n_ref:s+n_ref, t-n_ref:t+n_ref);
                    morph_matrix(s,t) =  max(temp(struct_indx));
                    %w1=[f(s-1,t-1)*w(1) f(s-1,t)*w(2) f(s-1,t+1)*w(3)␣
↪f(s,t-1)*w(4) f(s,t)*w(5) f(s,t+1)*w(6) f(s+1,t-1)*w(7) f(s+1,t)*w(8)␣
↪f(s+1,t+1)*w(9)];
                    %morph_matrix(s,t)=max(w1);
                end
            end

        elseif operation == 2
```

```
            %erosion

            for s=1+n_ref:x-n_ref
                for t=1+n_ref:y-n_ref
                    temp = gen_img(s-n_ref:s+n_ref, t-n_ref:t+n_ref);
                    morph_matrix(s,t) = min(temp(struct_indx));
                    %w1=[f(s-1,t-1)*w(1) f(s-1,t)*w(2) f(s-1,t+1)*w(3)␣
↪f(s,t-1)*w(4) f(s,t)*w(5) f(s,t+1)*w(6) f(s+1,t-1)*w(7) f(s+1,t)*w(8)␣
↪f(s+1,t+1)*w(9)];
                    %morph_matrix(s,t)=max(w1);
                end
            end
        end
    else
        if operation == 1
            %dilatacion

            for s=1+n_ref:x-n_ref
                for t=1+n_ref:y-n_ref
                    if gen_img(s,t) == 1
                        temp = gen_img(s-n_ref:s+n_ref, t-n_ref:t+n_ref);
                        temp(struct_indx) = 1;
                        current = morph_matrix(s-n_ref:s+n_ref, t-n_ref:
↪t+n_ref);
                        current = current + temp;
                        current = current ~= 0;
                        morph_matrix(s-n_ref:s+n_ref, t-n_ref:t+n_ref) =␣
↪current;
                    end
                end
            end

        elseif operation == 2
            %erosion

            for s=1+n_ref:x-n_ref
                for t=1+n_ref:y-n_ref
                    if gen_img(s,t) == 1
                        temp = gen_img(s-n_ref:s+n_ref, t-n_ref:t+n_ref);
                        if sum(sum(temp(struct_indx))) == struct_sum
                                morph_matrix(s,t) = 1;
                        end
                    end
                end
            end
        end
    end
```

```
    morph_matrix = morph_matrix(1+n_ref:end-n_ref, 1+n_ref:end-n_ref);
end
```

[84]:
```
function structural_element = gen_struct(radio, pre_define)

    % como usar esta funcion
    % marcar un radio, SIEMPRE IMPAR
    % seleccionar un tipo de estructura:
    % 1 Linea Vertical, 2 Linea horizonta, 3 Cuadrado, 4 Cruz, 5 Circulo, 6␣
 ↪Diamante

    structural_element = zeros(radio, radio);
    sub_rad = (radio - 1) / 2;
    switch pre_define
        case 1
            % linea vertical
            structural_element(:,1+sub_rad) = 1;
        case 2
            % linea horizontal
            structural_element(1+sub_rad, :) = 1;
        case 3
            % cuadrado
            structural_element(:,:) = 1;
        case 4
            % cruz
            structural_element(:, 1+sub_rad) = 1;
            structural_element(1+sub_rad, :) = 1;
        case 5
            % circulo

        case 6
            % diamante
            structural_element(1:1+sub_rad,(end - sub_rad):end) =␣
 ↪tril(ones(1+sub_rad));
            structural_element = structural_element + structural_element';
            structural_element = structural_element + rot90(structural_element);
            structural_element = structural_element ~= 0;
        otherwise
            structural_element = ones(radio, radio)
        end

end
```

[7]:
```
debug_DIL_STRUC = [0 1 0; 1 1 1; 0 1 0];
subplot(1,3,1); imshow(debug_DIL_STRUC);

debug_DIL = zeros(10,10);
```

```
debug_DIL(3,4:7) = 1;
debug_DIL(3:8,4) = 1;
debug_DIL(8,4:7) = 1;

subplot(1,3,2); imshow(debug_DIL);

debug_DIL = padding(debug_DIL, 2, size(debug_DIL_STRUC));
debug_DIL_RE = morphologic_converter(debug_DIL, 1, debug_DIL_STRUC, false);
subplot(1,3,3); imshow(debug_DIL_RE);
```

[8]:
```
debug_ERO_STRUC = [0 1 0; 1 1 1; 0 1 0];
subplot(1,3,1); imshow(debug_ERO_STRUC);

debug_ERO = zeros(10,10);
debug_ERO(3,4:6) = 1;
debug_ERO(4:6,3:7) = 1;
debug_ERO(7,4:6) = 1;

subplot(1,3,2); imshow(debug_ERO);
```

```
debug_ERO = padding(debug_ERO, 2, size(debug_ERO_STRUC));
debug_ERO_RE = morphologic_converter(debug_ERO, 2, debug_ERO_STRUC, false);
subplot(1,3,3); imshow(debug_ERO_RE);
```

[9]:
```
img =imread("../images/coins.png");
size(img)
```

```
ans =

   246   300
```

[85]:
```
struct = gen_struct(5, 4)
```

```
struct =

   0   0   1   0   0
   0   0   1   0   0
   1   1   1   1   1
   0   0   1   0   0
   0   0   1   0   0
```

[ ]:

[86]:
```
pre_img = padding(img, 1, size(struct));
size(pre_img)
imshow(pre_img)
```

```
ans =

   250   304
```

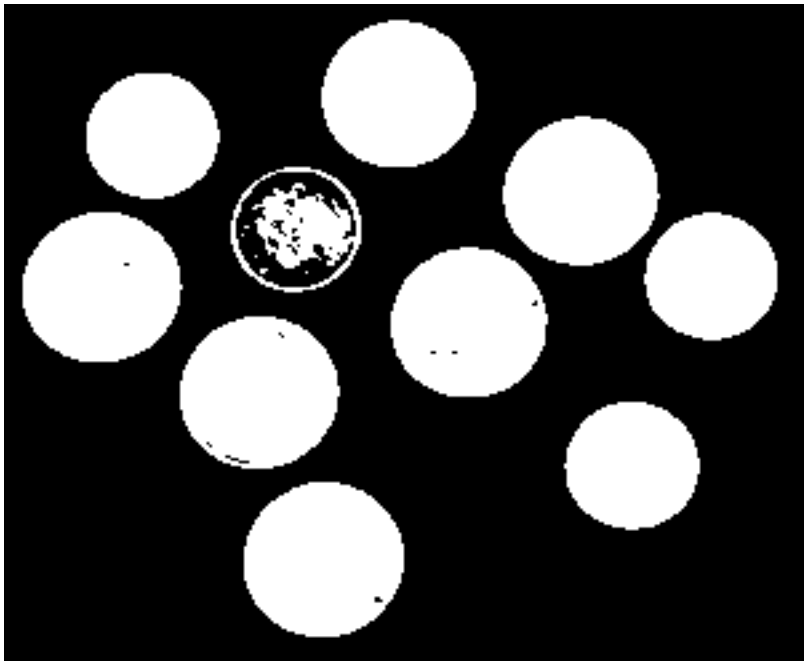
[87]:
```
morph_matrix  = morphologic_converter(pre_img, 1, struct, true);
imshow(morph_matrix)
```

[88]:
```
morph_matrix  = morphologic_converter(pre_img, 2, struct, true);
imshow(morph_matrix)
```

[ ]:
```

```

[89]:
```
img_bw = im2bw(img,graythresh(img));
% img_bw = img_bw ~= 1;
imshow(img_bw);
```



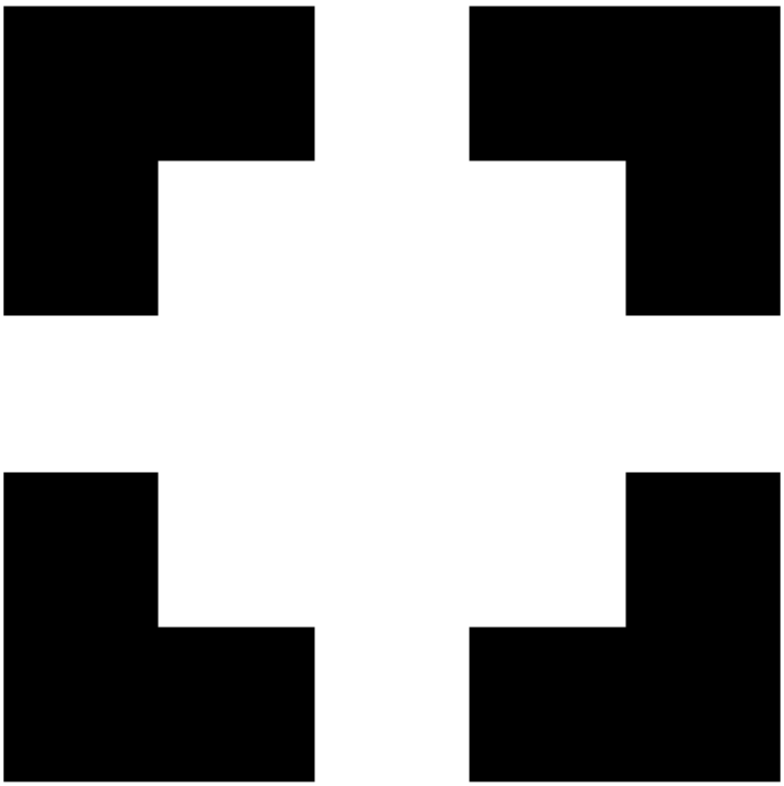
[90]:
```
pre_img_bw = padding(img_bw, 1, size(struct));
```

[91]:
```
struct = gen_struct(5,6)
```

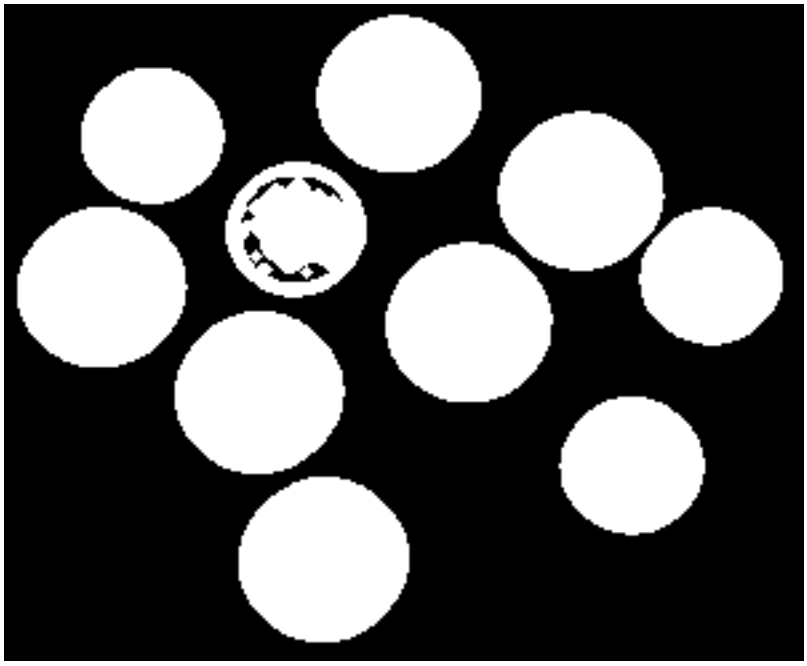
```
struct =

  0  0  1  0  0
  0  1  1  1  0
  1  1  1  1  1
  0  1  1  1  0
  0  0  1  0  0
```

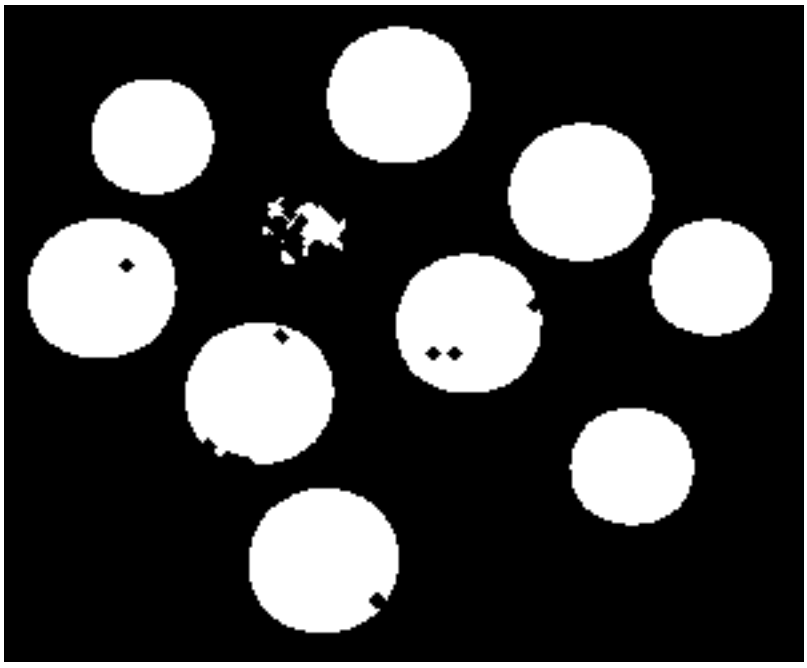
[92]:
```
imshow(struct)
```



[93]:
```
morph_matrix  = morphologic_converter(pre_img_bw, 1, struct, false);
imshow(morph_matrix)
```



[94]:
```
morph_matrix  = morphologic_converter(pre_img_bw, 2, struct, false);
imshow(morph_matrix)
```



[ ]:
```

```