Lab9-Final

September 17, 2020

[84]:
```
pkg load image
```

[85]:
```
% %Program No:5
% Write a histogram equalization function.

function histequal(f)

    g=histeq(f,256);

    imshow(f);
    figure, imhist(f);
    figure, imshow(g);
    figure, imhist(g);

end
```

[86]:
```
% Program No:6
% Write an M-function for performing local histogram equalization

function g = localhist(f)
    f=im2double(f);

    w=input('\nEnter the Neighborhood or Window size : ');
    k=input('\nEnter the value of the constant k (value should be between 0 and␣
 ↪1) : ');

    M=mean2(f);

    z=colfilt(f,[w w],'sliding',@std);
    m=colfilt(f,[w w],'sliding',@mean);

    A=k*M./z;
    g=A.*(f-m)+m;

%     imshow(f), figure, imshow(g);
end
```

[87]:
```
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
```

[88]:
```
function output = aMax(matrix_D)
    output = max(max(matrix_D));
end

function output = aSum(matrix_D)
```

```
    output = sum(sum(matrix_D));
end

function output = aMin(matrix_D)
    output = min(min(matrix_D));
end

function output = L2(glob_max)
    L = ceil(log2(glob_max + 1));
    output = pow2(L);
end
```

[89]:
```
function [output, spc_out] = histograma(imgData, ref_max)

    output = zeros(1,ref_max+1);
    spc_out = zeros(1,ref_max+1);

    prev = 1;
    prev_tmp = 0;
    for ii  = double(unique(imgData)')
        temp = aSum(imgData == ii);
        prev_tmp = prev_tmp + temp;

        output(ii+1) = temp;
        spc_out(prev:ii+1) = prev_tmp;

        prev = ii + 2;
        if prev > ref_max + 1
            prev = ref_max + 1;
        end
    end

    spc_out = cumsum(output);
end
```

[90]:
```
function equalized = contrastStretching(imgData, lazzo, verbose)
    imgData = double(imgData);
    ref_min = aMin(imgData);
    ref_max = aMax(imgData);

    new_ref_min = floor((1 - lazzo) * ref_min);
    new_ref_max = floor((1 + lazzo) * ref_max);

    if new_ref_min < 0
        new_ref_min = 0;
    elseif new_ref_min > 255
        new_ref_min = 255;
```

```
    end

    if new_ref_max < 0
        new_ref_max = 0;
    elseif new_ref_max > 255
        new_ref_max = 255;
    end

    if verbose
        disp('OLD RANGE')
        disp(ref_min)
        disp(ref_max)
        disp('NEW RANGE')
        disp(new_ref_min)
        disp(new_ref_max)
    end

    if (ref_max - ref_min) < 1e-16
        equalized = ones(size(imgData)) * ref_min;
    else
        equalized = (imgData - ref_min) / (ref_max - ref_min);
        equalized = equalized * (new_ref_max - new_ref_min);
        equalized = equalized + new_ref_min;
        equalized = uint8(floor(equalized));
    end
end
```

[91]:
```
function equalized = histEqualizer(imgData)
    imgData = double(imgData);

    [M, N] = size(imgData);
    equalized = zeros(M, N);

    glob_max = aMax(imgData);
    L_1 = L2(glob_max) - 1;

    [reg_hist, cum_hist] = histograma(imgData, L_1);
    p_i = double(cum_hist) / (M * N);

    for ii = unique(imgData)'
        new_val = L_1 * p_i(ii + 1);

        equalized(find(imgData == ii)) = new_val;
    end

    equalized = uint8(round(equalized));
end
```

[ ]:

[92]:
```
%
```

[ ]:

[93]:
```
function equalized = localeConstrastStreching(imgData, n_filter, lazzo, verbose)
    gen_img= padding(imgData, 2, [n_filter, n_filter]);
%     gen_img = uint8(gen_img);

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

    n_ref = (n_filter - 1) / 2;

    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);
            temp = contrastStretching(temp, lazzo, verbose);

            equalized(s,t) =  temp(n_ref+1, n_ref+1);
        end
    end
    equalized = equalized(1+n_ref:end-n_ref, 1+n_ref:end-n_ref);
    equalized = uint8(equalized);
end
```

[ ]:

[94]:
```
function equalized = histEqualizerSpecial(imgData)
    imgData = double(imgData);

    [M, N] = size(imgData);
    equalized = zeros(M, N);

    L_1 = 255;

    [reg_hist, cum_hist] = histograma(imgData, L_1);
    p_i = double(cum_hist) / (M * N);

    for ii = unique(imgData)'
        new_val = L_1 * p_i(ii + 1);

        equalized(find(imgData == ii)) = new_val;
    end

    equalized = uint8(round(equalized));
```

```
end
```

[95]:
```
function equalized = localeHistEqualizer(imgData, n_filter)
    gen_img= padding(imgData, 2, [n_filter, n_filter]);

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

    n_ref = (n_filter - 1) / 2;

    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);

%             disp(temp)
%             disp(temp(n_ref+1, n_ref+1))

            temp = histEqualizerSpecial(temp);

%             disp(temp)
%             disp(temp(n_ref+1, n_ref+1))
%             disp('---')

            equalized(s,t) =  temp(n_ref+1, n_ref+1);
        end
    end
    equalized = equalized(1+n_ref:end-n_ref, 1+n_ref:end-n_ref);
    equalized = uint8(equalized);
end
```

[ ]:
```
```

[96]:
```
%
```

[ ]:
```
```

[103]:
```
BASE_PATH = './ImagenesContraste/';
FILES_SETS = {
   {
    "barbaraD.png"
   ,"barbaraL.png"
   ,"barbara.png"
   }
   ,
   {
    "hands1D.png"
   ,"hands1L.png"
```

```
   ,"hands1.png"
   }
   ,
   {
    "lakeD.png"
   ,"lakeL.png"
   ,"lake.png"
   }
   ,
   {
    "LenaDark1.png"
   ,"LenaDark2.png"
   ,"LenaLight1.png"
   ,"LenaLight2.png"
   ,"lena.png"
   }
   ,
   {
    "peppersD.png"
   ,"peppersL.png"
   ,"peppers.png"
   }
   ,
   {"pout.png"}
};
```

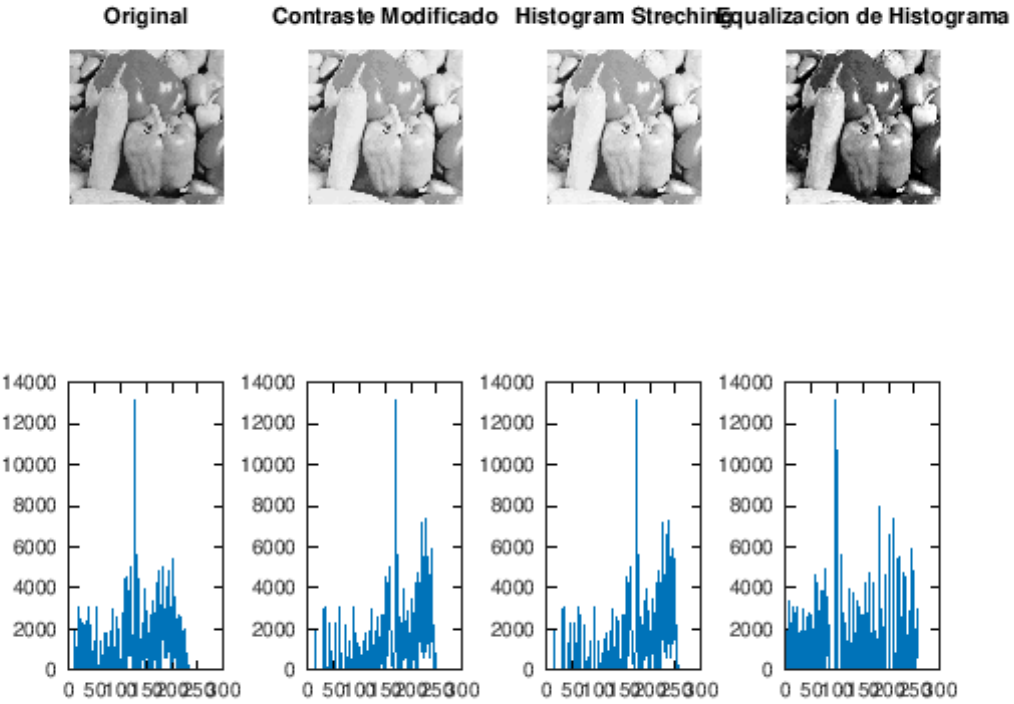
[ ]:

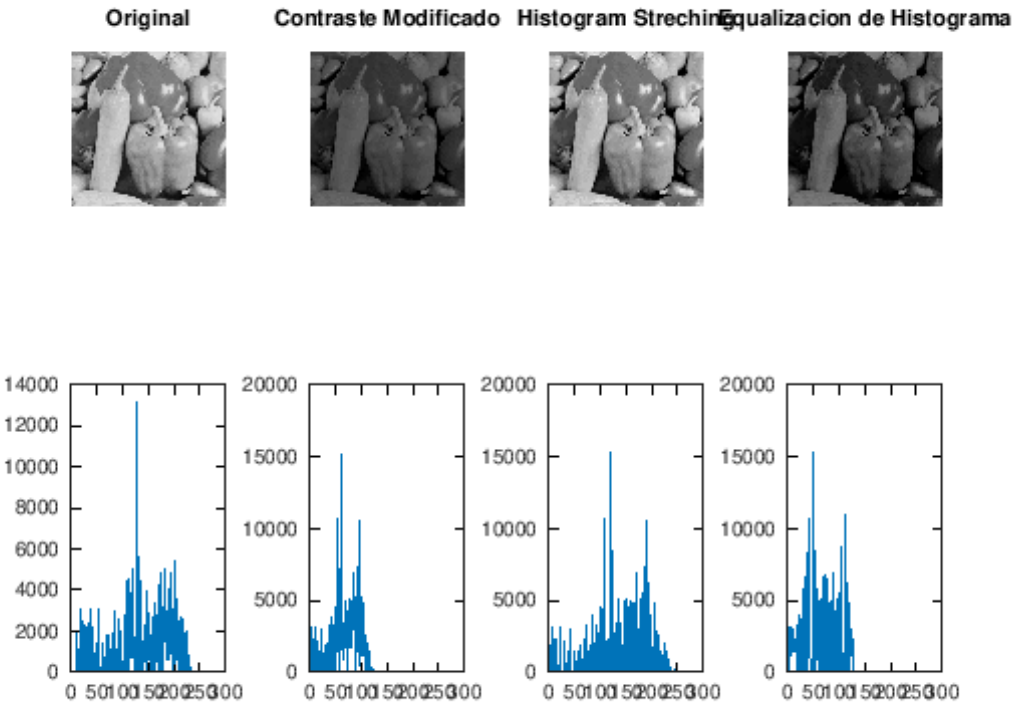
[143]:
```
% meters = []
id_fig = 1;

for ii_file_set = 1:size(FILES_SETS)(1)
    alt_current_set = FILES_SETS{ii_file_set};

    ref_img = alt_current_set{size(alt_current_set)(1)};
    ref_img = strcat(BASE_PATH, ref_img);
    ref_img = imread(ref_img);
    if size(size(ref_img))(2) == 3
        ref_img = rgb2gray(ref_img);
    end

    for jj_file_set = 1:size(alt_current_set)(1) - 1
        current_img_path = alt_current_set{jj_file_set};

        current_path = strcat(BASE_PATH, current_img_path);
        im_ref = imread(current_path);
```

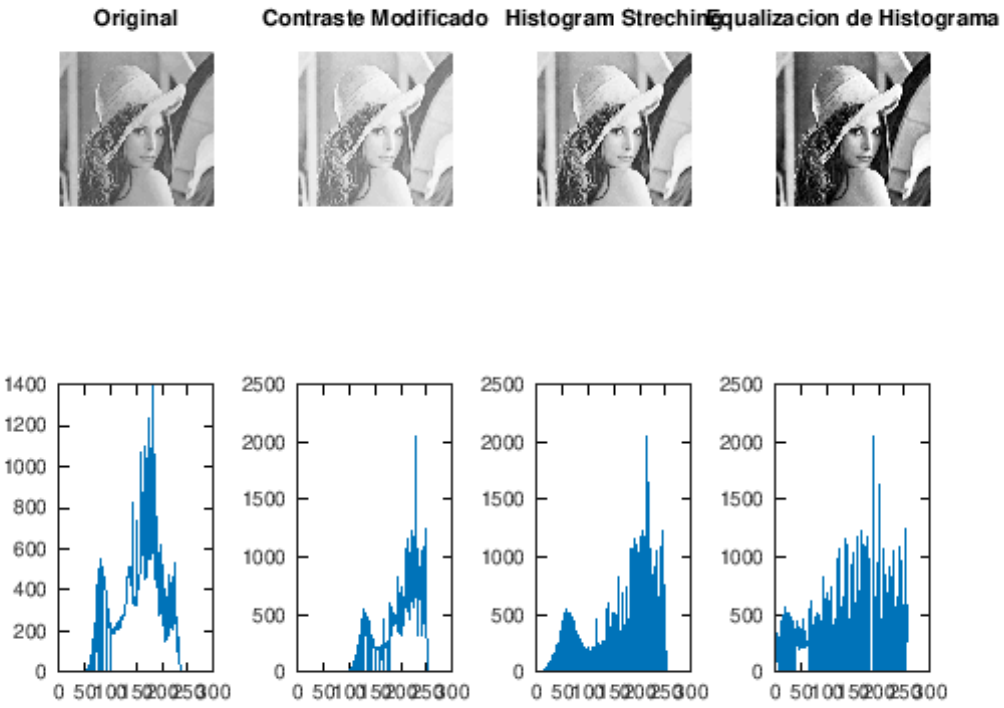
```
        if size(size(im_ref))(2) == 3
            im_ref = rgb2gray(im_ref);
        end

        %
        curr_stret = contrastStretching(im_ref, 1, false);
        curr_equal = histEqualizer(im_ref);

        immse_stret = immse(curr_stret, im_ref);
        psnr_stret = psnr(curr_stret, im_ref);
%         ssim_stret = ssim(curr_stret, im_ref);

        immse_equal = immse(curr_equal, im_ref);
        psnr_equal = psnr(curr_equal, im_ref);
%         ssim_equal = ssim(curr_equal, im_ref);

        line_stat = [
        immse_stret,
        psnr_stret,
%         ssim_stret,
        immse_equal,
        psnr_equal,
%         ssim_equal
        ]';

        meters = [meters; line_stat];
        %

        figure;

        subplot(2,4,1); imshow(ref_img); title("Original");
        subplot(2,4,2); imshow(im_ref); title("Contraste Modificado");
        subplot(2,4,3); imshow(curr_stret); title("Histogram Streching");
        subplot(2,4,4); imshow(curr_equal); title("Equalizacion de Histograma");

        subplot(2,4,5); plot(histograma(ref_img, 255));
        subplot(2,4,6); plot(histograma(im_ref, 255));
        subplot(2,4,7); plot(histograma(curr_stret, 255));
        subplot(2,4,8); plot(histograma(curr_equal, 255));

        saveas(id_fig, strjoin({'./ImagenesContraste/', 'output/
 ↪GEN_EQ_',current_img_path}, ''))
    end
end
```

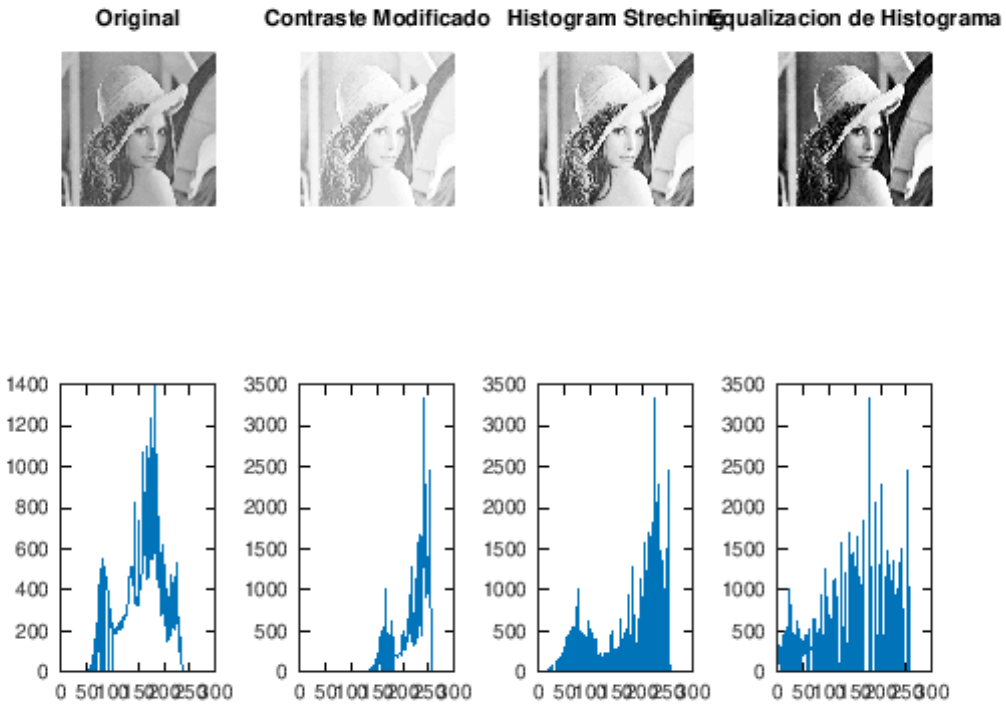
```
DEBUG: FC_WEIGHT didn't match
DEBUG: FC_WEIGHT didn't match
```

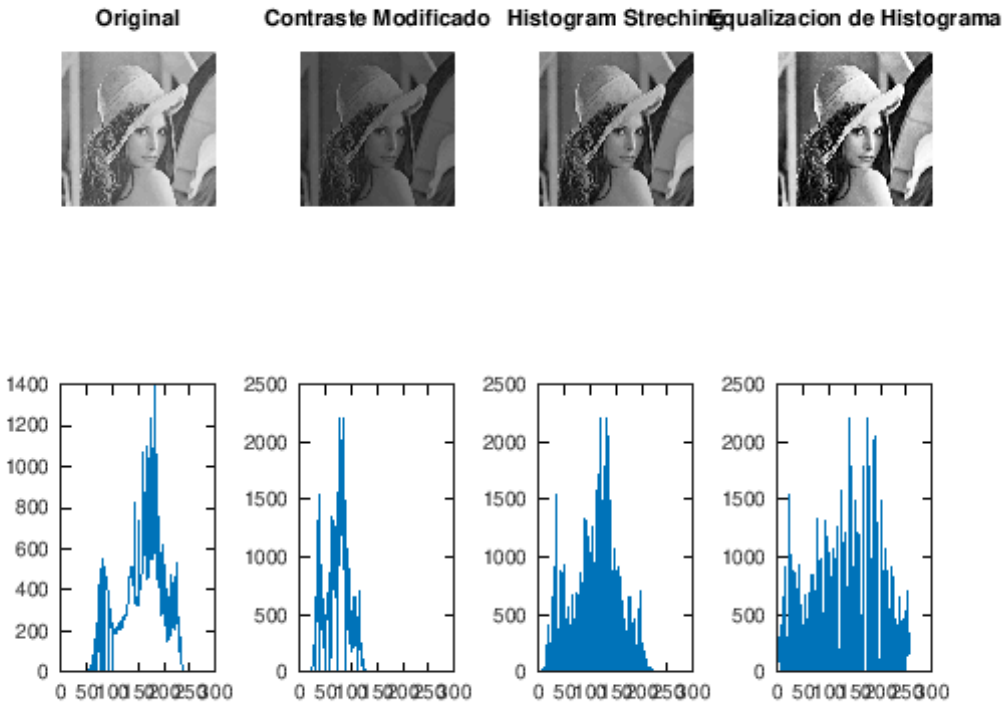
```
DEBUG: FC_WEIGHT didn't match
DEBUG: FC_WEIGHT didn't match
DEBUG: FC_WEIGHT didn't match
DEBUG: FC_WEIGHT didn't match
DEBUG: FC_WEIGHT didn't match
DEBUG: FC_WEIGHT didn't match
DEBUG: FC_WEIGHT didn't match
DEBUG: FC_WEIGHT didn't match
DEBUG: FC_WEIGHT didn't match
DEBUG: FC_WEIGHT didn't match
```

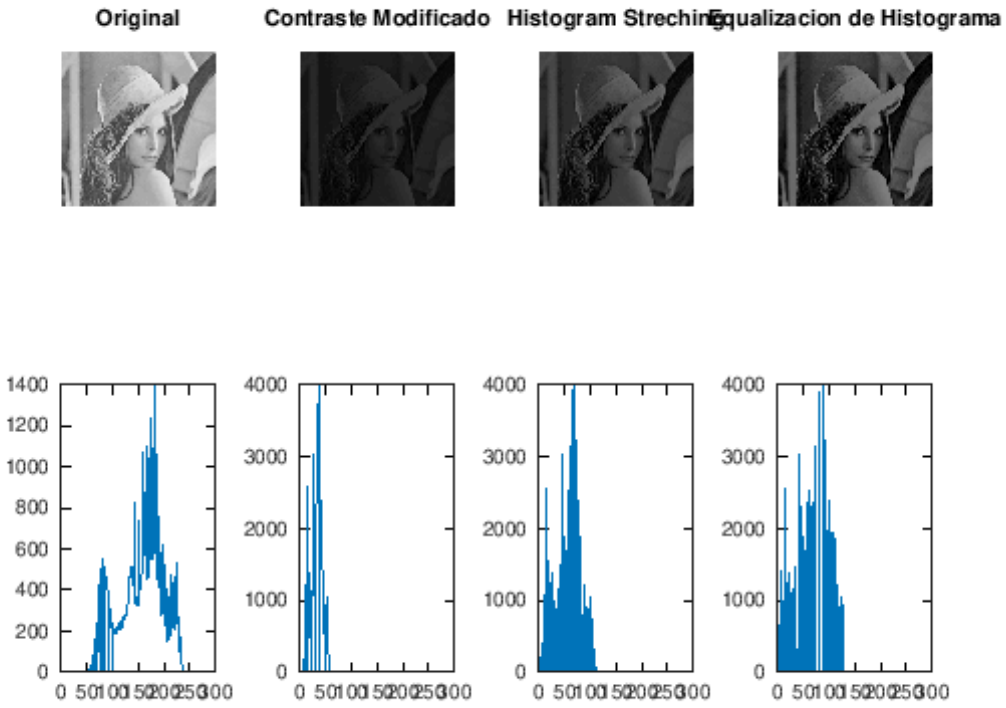


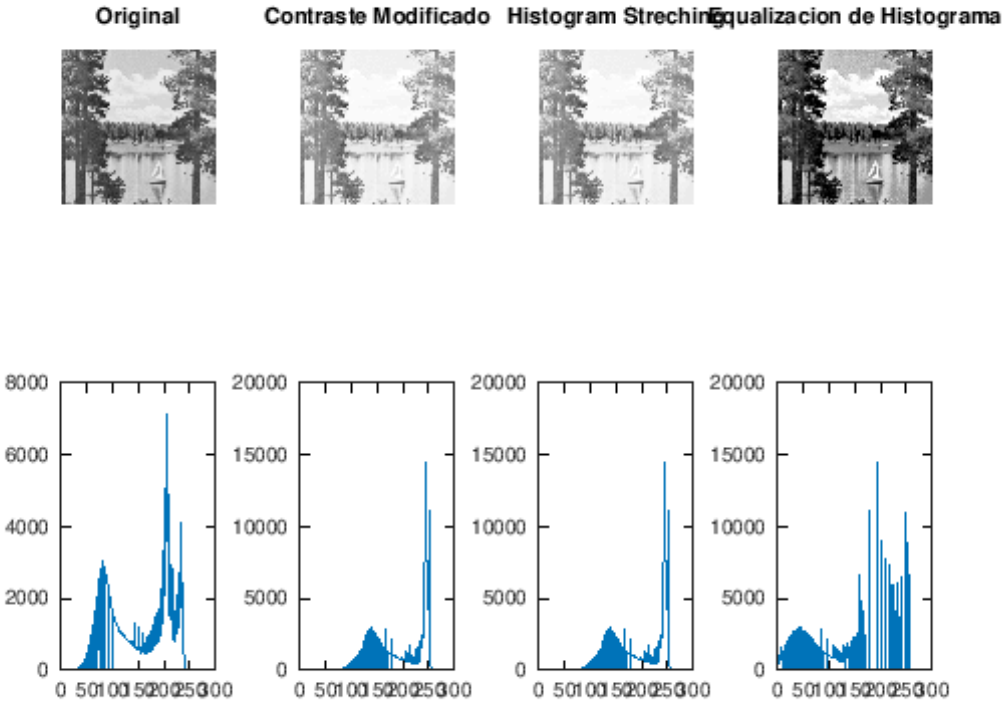

Original
Contraste Modificado
Histogram Strechin
Equalizacion de Histograma
14000
12000
10000
8000
6000
4000
2000
0
20000
15000
10000
5000
0
0 50 100 150 200 250 300

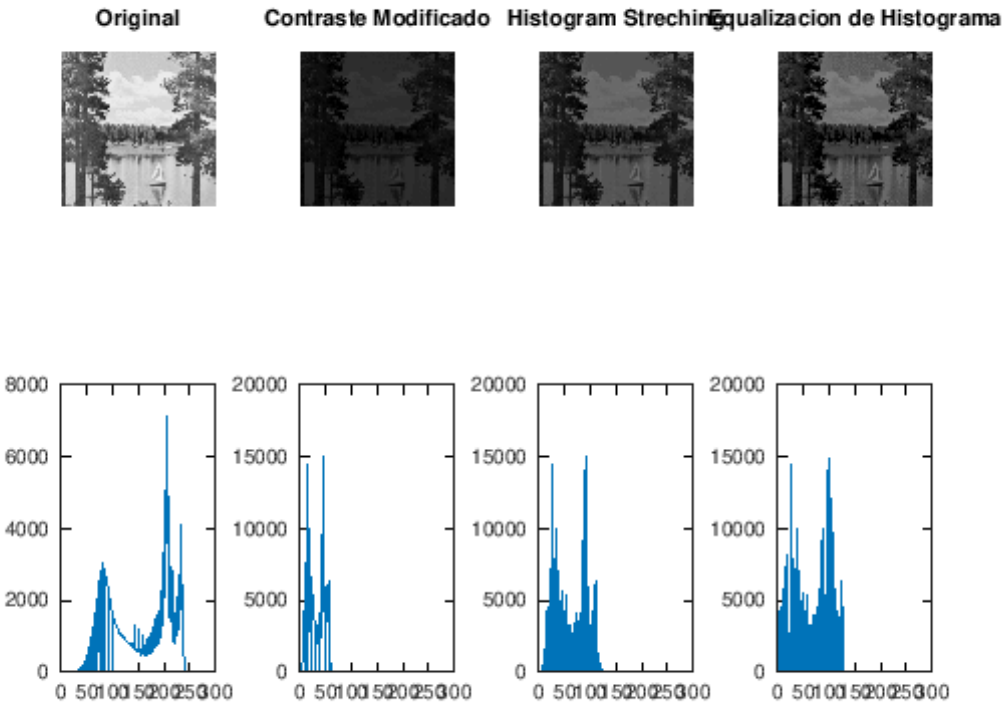

Original
Contraste Modificado
Histogram Strechin
Equalizacion de Histograma
1400
1200
1000
800
600
400
200
0
2500
2000
1500
1000
500
0
0 50 100 150 200 250 300

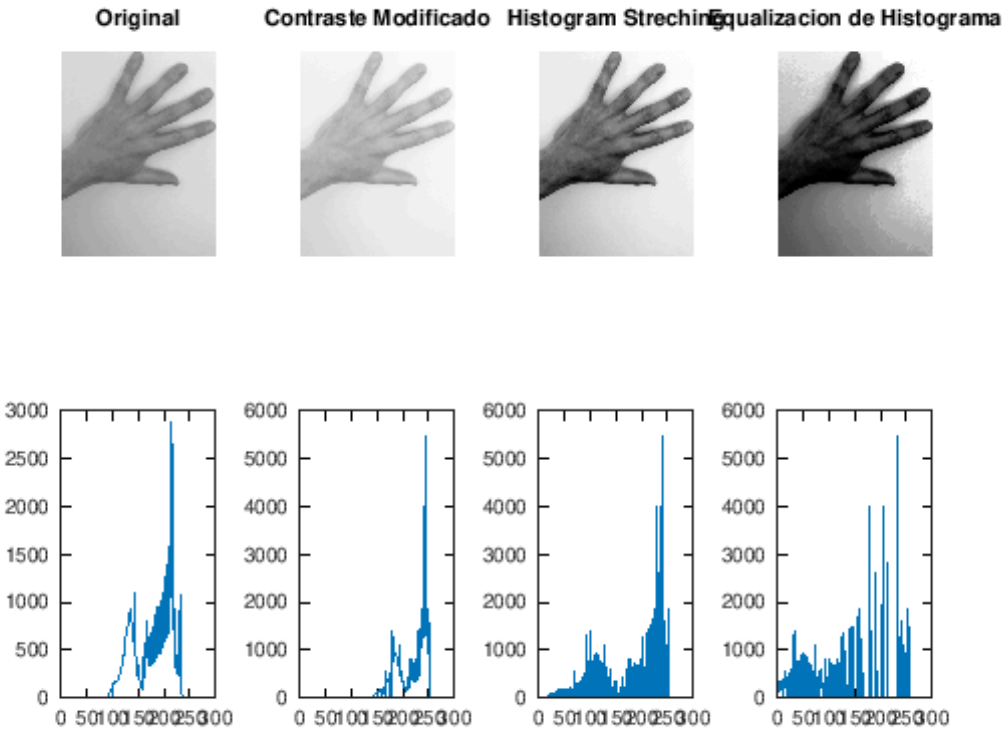

Original
Contraste Modificado
Histogram Strechin
Equalizacion de Histograma
1400
1200
1000
800
600
400
200
0
3500
3000
2500
2000
1500
1000
500
0
0 50 100 150 200 250 300

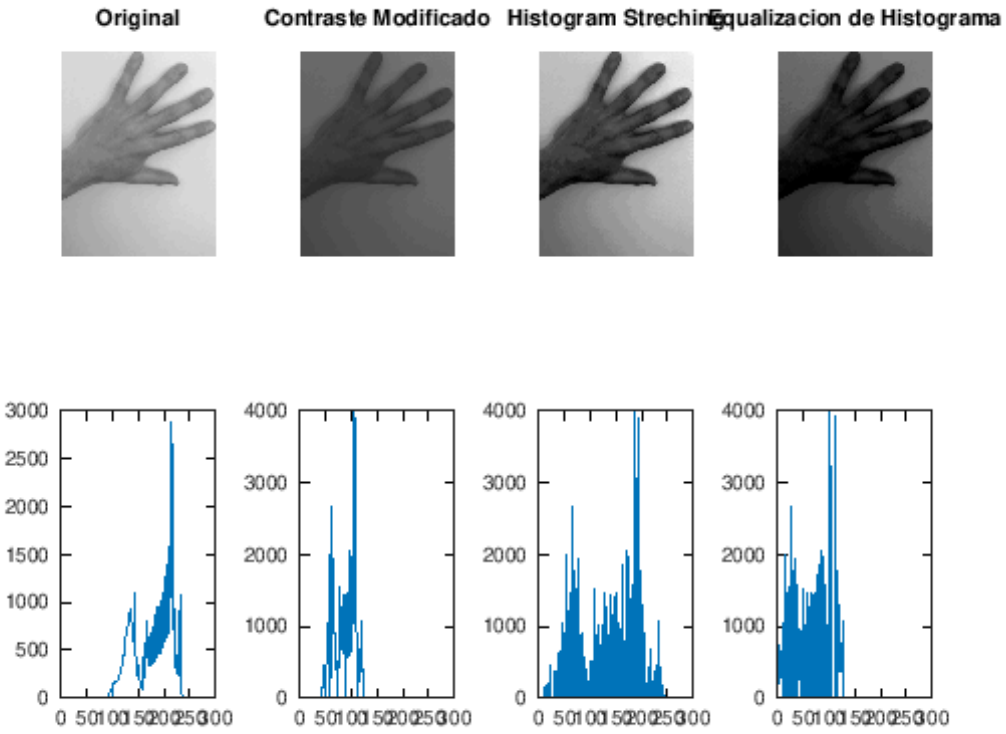

Original
Contraste Modificado
Histogram Strechin
Equalizacion de Histograma
1400
1200
1000
800
600
400
200
0
2500
2000
1500
1000
500
0
0 50 100 150 200 250 300

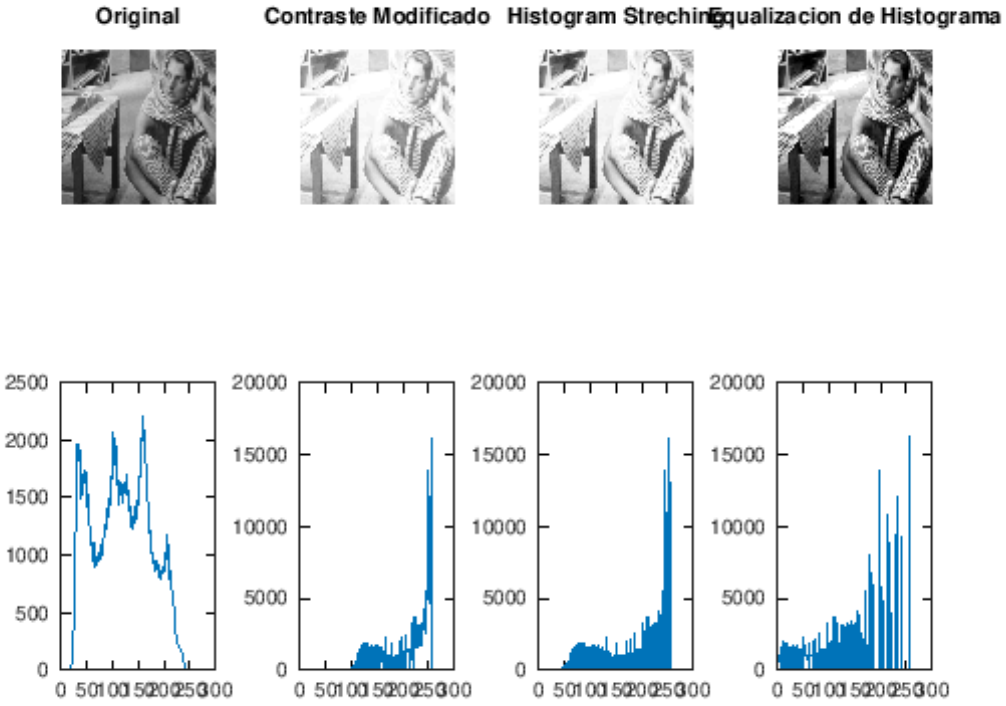

Original
Contraste Modificado
Histogram Strechin
Equalizacion de Histograma
1400
1200
1000
800
600
400
200
0
4000
3000
2000
1000
0
0 50 100 150 200 250 300

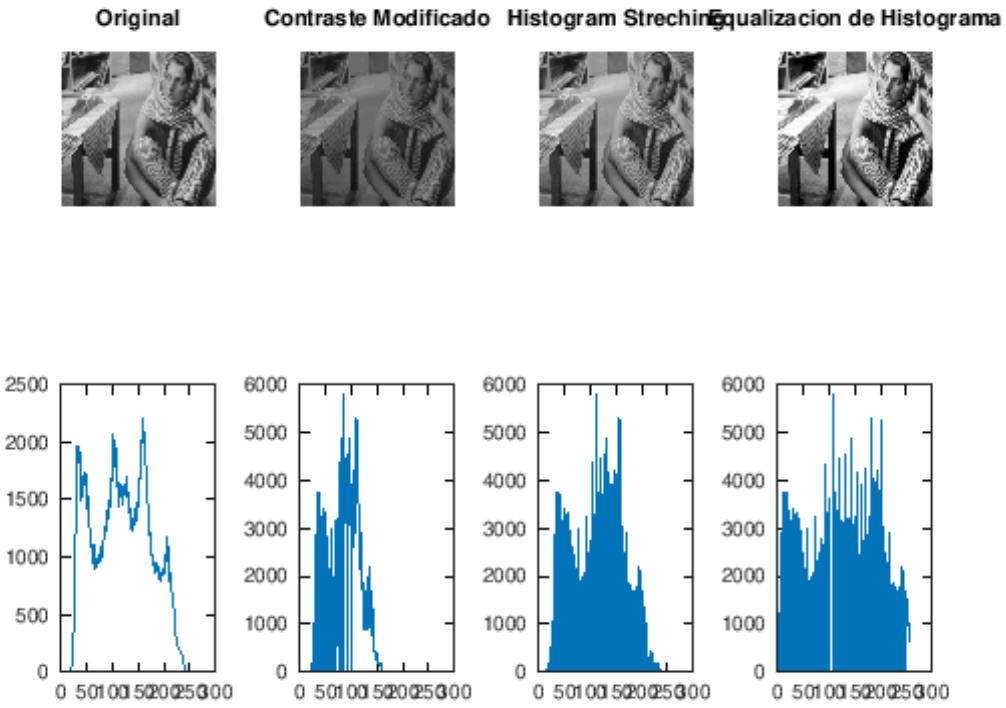

Original
Contraste Modificado
Histogram Strechin
Equalizacion de Histograma
8000
6000
4000
2000
0
20000
15000
10000
5000
0
0 50 100 150 200 250 300


Original
Contraste Modificado
Histogram Strechin
Equalizacion de Histograma
8000
6000
4000
2000
0
20000
15000
10000
5000
0
0 50 100 150 200 250 300


Original
Contraste Modificado
Histogram Strechin
Equalizacion de Histograma
3000
2500
2000
1500
1000
500
0
6000
5000
4000
3000
2000
1000
0
0 50 100 150 200 250 300


Original
Contraste Modificado
Histogram Strechin
Equalizacion de Histograma
3000
2500
2000
1500
1000
500
0
4000
3000
2000
1000
0
0 50 100 150 200 250 300


Original
Contraste Modificado
Histogram Strechin
Equalizacion de Histograma
2500
2000
1500
1000
500
0
20000
15000
10000
5000
0
0 50 100 150 200 250 300



[150]:
```
csvwrite("./data.csv", meters);
```

[ ]:
```

```