

Contents

- 1. Write a matlab program to implement contrast stretching of an image.
- CASE 1
- CASE 2
- CASE 3
- 2. Demonstrate Power Law transformation of an image.
- CASE 1
- CASE 2

1. Write a matlab program to implement contrast stretching of an image.

CASE 1

stretchlim - Find limits to contrast stretch image

```
% lowhigh = stretchlim(I) computes the lower and upper limits that can be used for contrast stretching grayscale or RGB image I.

I = imread('pout.tif');

% The limits are returned in lowhigh. By default, the limits specify the bottom 1% and the top 1% of all pixel values.
```

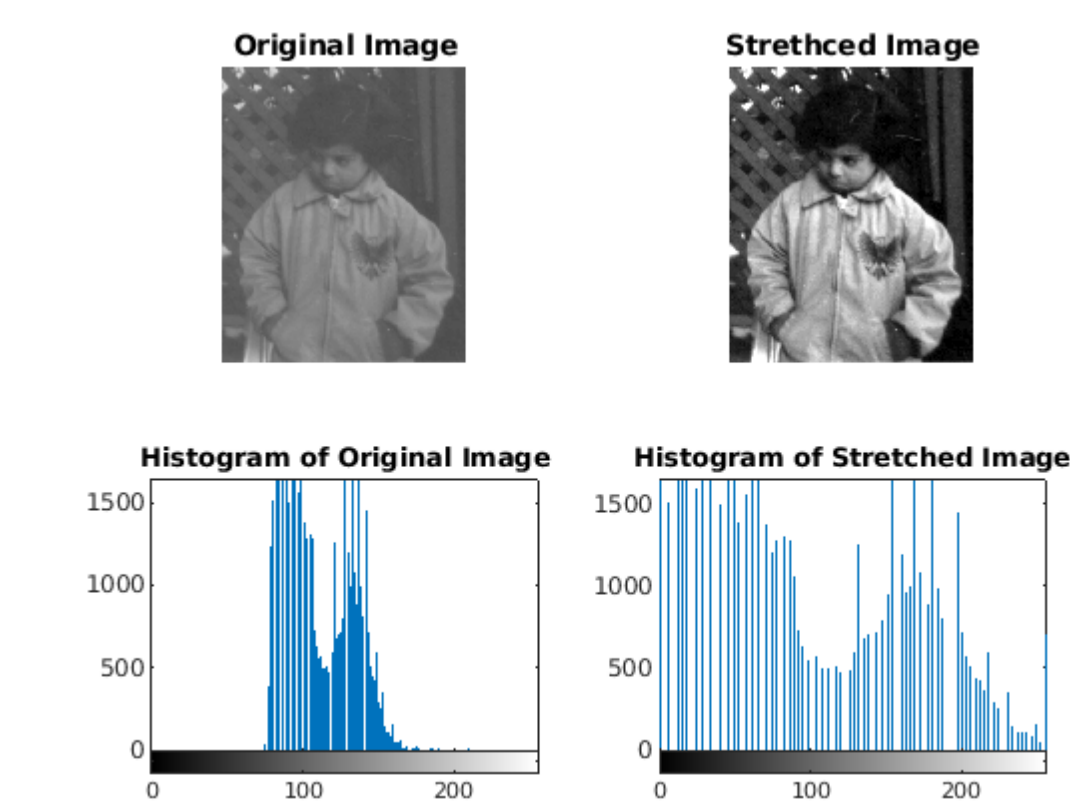
```
% lowhigh = stretchlim(I,Tol) specifies the fraction, Tol, of the image to saturate at low and high pixel values.

J = imadjust(I,stretchlim(I),[]);
```

Plotting

```
subplot(2,2,1), imshow(I), title('Original Image');
subplot(2,2,2), imshow(J), title('Strethced Image');

subplot(2,2,3), imhist(I), title('Histogram of Original Image');
subplot(2,2,4), imhist(J), title('Histogram of Stretched Image');
```



CASE 2

Contrast Stretching in Image Processing using Matlab Functions

```
image = imread('pout.tif');
stretched_Image = imadjust(image, stretchlim(image, [0.05, 0.95]),[]);
```

Plotting

```
subplot(2,2,1), imshow(image), title('Original Image');
subplot(2,2,2), imshow(stretched_Image), title('Strethced Image');
subplot(2,2,3), imhist(image), title('Histogram of Original Image');
subplot(2,2,4), imhist(stretched_Image), title('Histogram of Stretched Image');
```

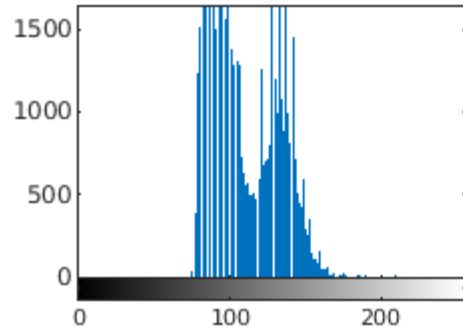
Original Image



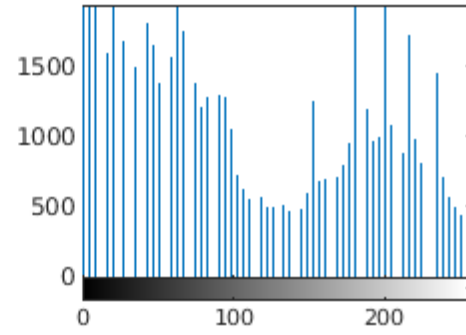
Strethced Image



Histogram of Original Image



Histogram of Stretched Image



CASE 3

Contrast Stretching in Image Processing using Manual Formula

```
pout_image = imread('pout.tif');  
[m,n] = size(pout_image);
```

Getting lower and upper limits that can be used for contrast stretching grayscale or RGB image

```
low_limit = 5; % input('Enter the Lower limit: ');  
high_limit = 230; % input('Enter the Upper Limit: ');
```

```
lowest_pixel = min(min(pout_image)); % lowest pixel  
highest_pixel = max(max(pout_image)); % highest pixel
```

```
for i = 1:m  
    for j = 1:n  
        array(i,j) = ((pout_image(i,j)-lowest_pixel) * ((high_limit-low_limit)/(highest_pixel-lowest_pixel))) + low_limit;  
    end  
end
```

Plotting

```
subplot(2,2,1), imshow(pout_image), title('Original Image');  
subplot(2,2,2), imshow(array), title('Strethced Image');  
subplot(2,2,3), imhist(pout_image), title('Histogram of Original Image');  
subplot(2,2,4), imhist(array), title('Histogram of Stretched Image');
```

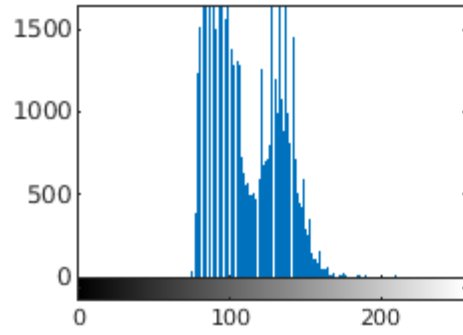
Original Image



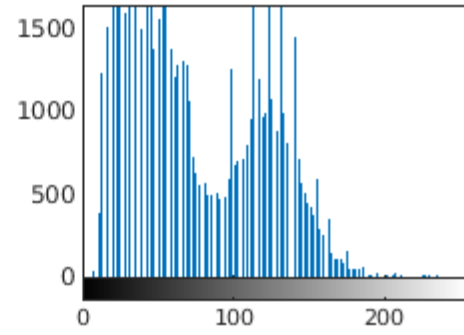
Stretched Image



Histogram of Original Image



Histogram of Stretched Image



2. Demonstrate Power Law transformation of an image.

```
%for gamma value less than 1 we get Bright image  
%for gamma value greater than 1 we get Dark image
```

CASE 1

```
itemp = imread('cameraman.tif'); % read the image  
r = double(itemp)/255; % normalise the image  
c = 1; % constant  
  
gamma = 0.4;  
gamma1 = 3;  
s = c*(r).^gamma; % formula to implement power law transformation  
s1 = c*(r).^gamma1;
```

Plotting

```
subplot(1,3,1),imshow(uint8(itemp)),title('Original ');  
subplot(1,3,2),imshow(s),title('Power Law');  
subplot(1,3,3),imshow(s1),title('Transformed Image');
```

Original



Power Law



Transformed Image



CASE 2

```
cameraman = imread('cameraman.tif');  
[m,n] = size(cameraman);  
  
Constand = 1;  
Gamme = 2;  
for i = 1:m  
    for j = 1:n  
        plt(i,j) = Constand * (cameraman(i,j)^Gamme);
```

```
end  
end
```

Plotting

```
subplot(2,2,1), imshow(cameraman), title('Original Image');  
subplot(2,2,2), imshow(plt), title('PLT Image');  
subplot(2,2,3), imhist(cameraman), title('Histogram of Original Image');  
subplot(2,2,4), imhist(cameraman), title('Histogram of PLT Image');
```

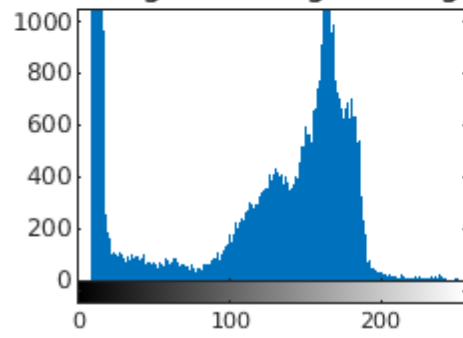
Original Image



PLT Image



Histogram of Original Image



Histogram of PLT Image

