Batch: A-4 **Roll No.:** 16010122151

Experiment No.

Grade: AA / AB / BB / BC / CC / CD /DD

Signature of the Staff In-charge with date

Title: Implement contrast stretching of a digital image.

Objective: To learn & understand contrast stretching.

Expected Outcome of Experiment:

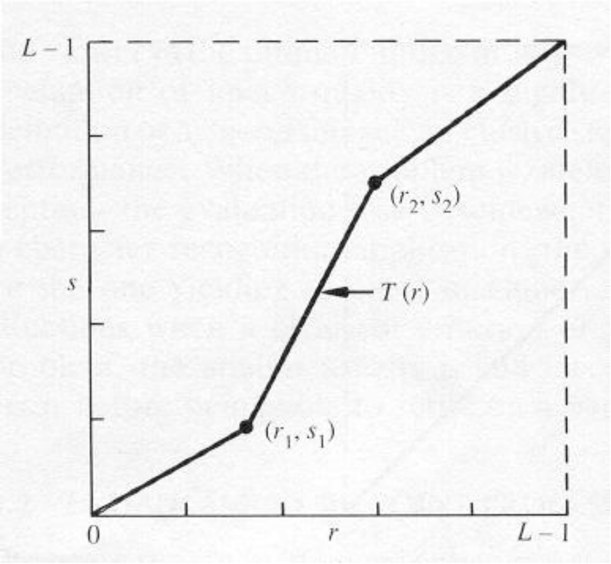
CO	Outcome
CO4	Design & implement algorithms for digital image enhancement, segmentation & restoration.

Books/ Journals/ Websites referred:

1. http://www.mathworks.com/support/
2. www.math.mtu.edu/~msgocken/intro/intro.html.
3. R. C.Gonsales R.E.Woods, “Digital Image Processing”, Second edition, Pearson Education
4. S.Jayaraman, S Esakkirajan, T Veerakumar “Digital Image Processing “Mc Graw Hill.
5. S.Sridhar,”Digital Image processing”, oxford university press, 1st edition."

Pre Lab/ Prior Concepts:

Contrast stretching (often called normalization) is a simple image enhancement technique that attempts to improve the contrast in an image by `stretching' the range of intensity values it contains to span a desired range of values, *e.g.* the the full range of pixel values that the image type concerned allows. It differs from the more sophisticated histogram equalization in that it can only apply a *linear* scaling function to the image pixel values. As a result the `enhancement' is less harsh.



The locations of (r_1,s_1) and (r_2,s_2) control the shape of the transformation function.

- If $r_1= s_1$ and $r_2= s_2$ the transformation is a linear function and produces no changes.
- If $r_1=r_2$, $s_1=0$ and $s_2=L-1$, the transformation becomes a thresholding function that creates a binary image.
- Intermediate values of (r_1,s_1) and (r_2,s_2) produce various degrees of spread in the gray levels of the output image, thus affecting its contrast.

Generally, $r_1 \leq r_2$ and $s_1 \leq s_2$ is assumed.

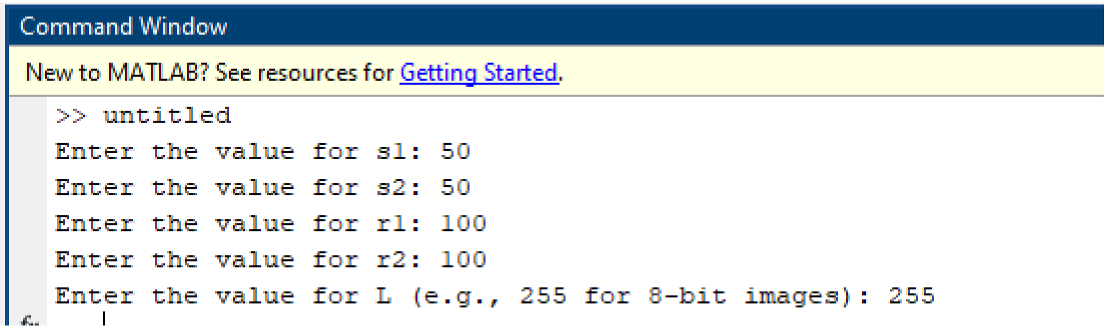
Implementation steps with screenshots:

```
img = imread('cosmos.bmp');
```

```
if size(img, 3) == 3

   img = rgb2gray(img);

end

min_val = double(min(img(:)));

max_val = double(max(img(:)));

% User-defined inputs for stretching parameters

s_low = input('Enter the value for s1: ');

s_high = input('Enter the value for s2: ');

r_low = input('Enter the value for r1: ');

r_high = input('Enter the value for r2: ');

bit_depth = input('Enter the value for L (e.g., 255 for 8-bit images):
');

scale1 = s_low / r_low;

scale2 = (s_high - s_low) / (r_high - r_low);

scale3 = ((bit_depth - 1) - s_high) / ((bit_depth - 1) - r_high);

% Applying contrast stretching

stretched_img1 = double(img);

stretched_img1(img <= r_low) = scale1 * img(img <= r_low);

stretched_img1(img > r_low & img <= r_high) = scale2 * (img(img >
r_low & img <= r_high) - r_low) + s_low;

stretched_img1(img > r_high) = scale3 * (img(img > r_high) - r_high) +
s_high;

stretched_img1 = uint8(stretched_img1);

% Generating second variation with modified parameters

s_low2 = s_low + 20;

s_high2 = s_high + 20;

r_low2 = r_low + 20;

r_high2 = r_high + 20;

scale1_2 = s_low2 / r_low2;

scale2_2 = (s_high2 - s_low2) / (r_high2 - r_low2);
```

```
scale3_2 = ((bit_depth - 1) - s_high2) / ((bit_depth - 1) - r_high2);

stretched_img2 = double(img);

stretched_img2(img <= r_low2) = scale1_2 * img(img <= r_low2);

stretched_img2(img > r_low2 & img <= r_high2) = scale2_2 * (img(img >
r_low2 & img <= r_high2) - r_low2) + s_low2;

stretched_img2(img > r_high2) = scale3_2 * (img(img > r_high2) -
r_high2) + s_high2;

stretched_img2 = uint8(stretched_img2);

% Generating third variation with further adjustments

s_low3 = s_low + 40;

s_high3 = s_high + 40;

r_low3 = r_low + 40;

r_high3 = r_high + 40;

scale1_3 = s_low3 / r_low3;

scale2_3 = (s_high3 - s_low3) / (r_high3 - r_low3);

scale3_3 = ((bit_depth - 1) - s_high3) / ((bit_depth - 1) - r_high3);

stretched_img3 = double(img);

stretched_img3(img <= r_low3) = scale1_3 * img(img <= r_low3);

stretched_img3(img > r_low3 & img <= r_high3) = scale2_3 * (img(img >
r_low3 & img <= r_high3) - r_low3) + s_low3;

stretched_img3(img > r_high3) = scale3_3 * (img(img > r_high3) -
r_high3) + s_high3;

stretched_img3 = uint8(stretched_img3);

% Displaying results

figure;

subplot(2, 2, 1), imshow(img);

title('Original Image');

subplot(2, 2, 2), imshow(stretched_img1);

title(sprintf('Contrast Stretched 1 (r1=%.2f, s1=%.2f, r2=%.2f,
s2=%.2f)', r_low, s_low, r_high, s_high));
```

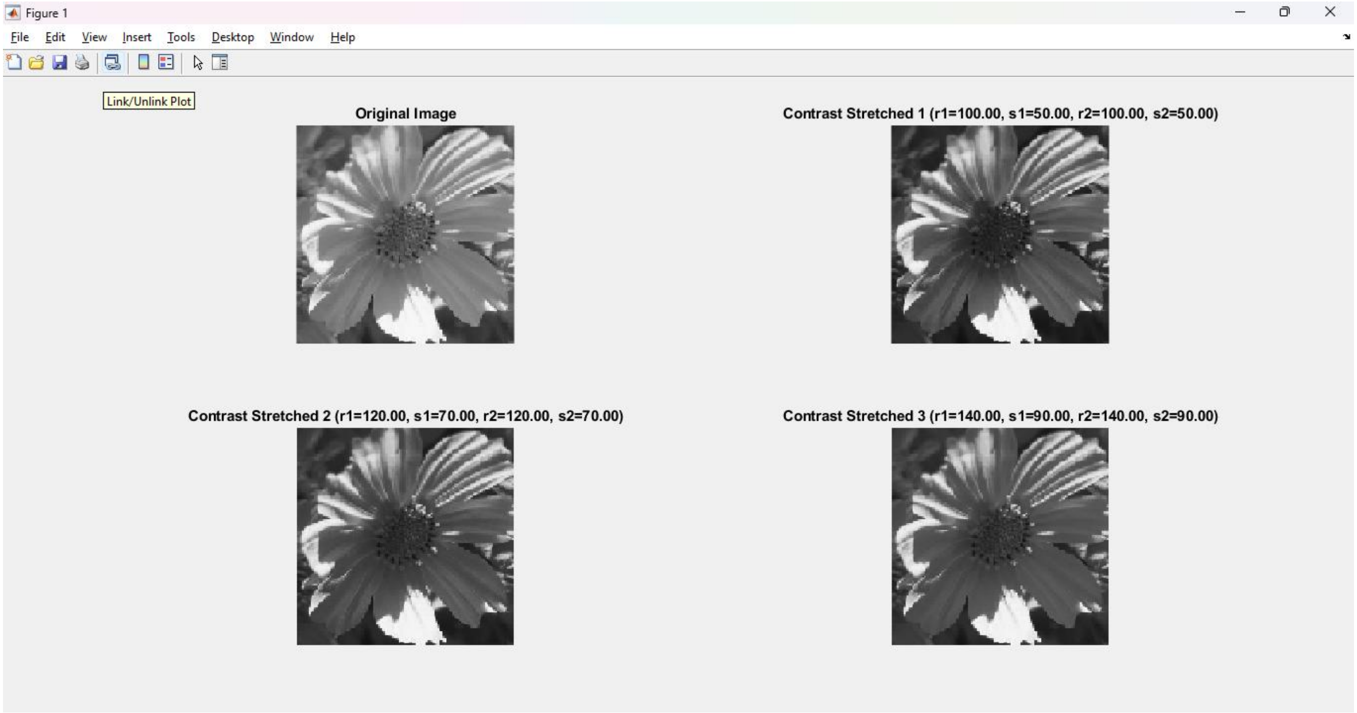
```
subplot(2, 2, 3), imshow(stretched_img2);

title(sprintf('Contrast Stretched 2 (r1=%.2f, s1=%.2f, r2=%.2f,
s2=%.2f)', r_low2, s_low2, r_high2, s_high2));

subplot(2, 2, 4), imshow(stretched_img3);

title(sprintf('Contrast Stretched 3 (r1=%.2f, s1=%.2f, r2=%.2f,
s2=%.2f)', r_low3, s_low3, r_high3, s_high3));
```



Command Window

New to MATLAB? See resources for Getting Started.

```
>> untitled
Enter the value for s1: 50
Enter the value for s2: 50
Enter the value for r1: 100
Enter the value for r2: 100
Enter the value for L (e.g., 255 for 8-bit images): 255
```



Original Image

Contrast Stretched 1 (r1=100.00, s1=50.00, r2=100.00, s2=50.00)

Contrast Stretched 2 (r1=120.00, s1=70.00, r2=120.00, s2=70.00)

Contrast Stretched 3 (r1=140.00, s1=90.00, r2=140.00, s2=90.00)

Conclusion:- Learned Contrast, Stretching on an image with varying parameters, displaying the original image alongside three progressively contrast-stretched versions.

Date: 12/03/2025 **Signature of faculty in-charge**

Post Lab Descriptive Questions

1. Thresholding function in contrast stretching creates
   a) binary image
   b) high quality image
   c) enhanced image
   d) low quality image

2. When is the contrast stretching transformation a linear function, for r and s as gray-value of image before and after processing respectively?
   a) r1 = s1 and r2 = s2
   b) r1 = r2, s1 = 0 and s2 = L – 1, L is the max gray value allowed
   c) r1 = 1 and r2 = 0
   d) None of the mentioned

3. Which gray-level transformation increase the dynamic range of gray-level in the image?

   a) Power-law transformations
   b) Negative transformations
   c) Contrast stretching
   d) None of the mentioned

4. When is the contrast stretching transformation a thresholding function, for r and s as gray-value of image before and after processing respectively?

a) r1 = s1 and r2 = s2

b) r1 = r2, s1 = 0 and s2 = L – 1, L is the max gray value allowed

c) r1 = 1 and r2 = 0

d) None of the mentioned

5. What condition prevents the intensity artifacts to be created while processing with contrast stretching, if r and s are gray-values of image before and after processing respectively?

a) r1 = s1 and r2 = s2

b) r1 = r2, s1 = 0 and s2 = L – 1, L is the max gray value allowed

c) r1 = 1 and r2 = 0

d) r1 ≤ r2 and s1 ≤ s2