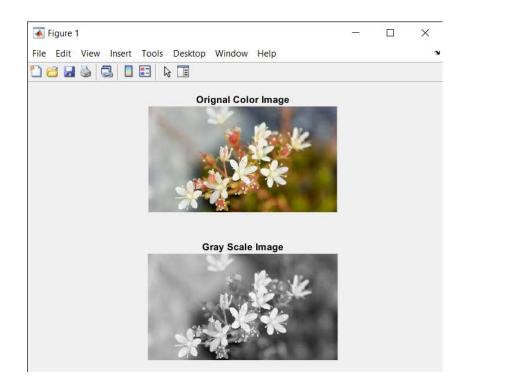
## PRACTICAL: 2

# AIM-1: Read your photo. Convert it into gray scale. Inspect the impact of sampling and quantization.

### **CODE:**

```
Editor - D:\Sem-6\CV\Practical-1\colorToGrayImage.m
   colorToGrayImage.m X negativeImage.m X contrastStreching.m X noiseImage.m X SmoothingFilter.n
       clear all;
 2 -
       close all;
 3 -
       clc;
       % reading the RGB file into the Matlab environment
       img = imread('colorImage.jpg');
 7
       % displaying the RGB image
       subplot(2, 1, 1);
9 -
       imshow(img);
       title('Orignal Color Image');
10 -
11
       %converting into gray image using rgb2gray function
12
13 -
       grayImg = rgb2gray(img);
14
15
       % displaying the gray scale image
       subplot(2, 1, 2);
16 -
17 -
       imshow(grayImg);
       title('Gray Scale Image');
18 -
```

### **OUTPUT:**

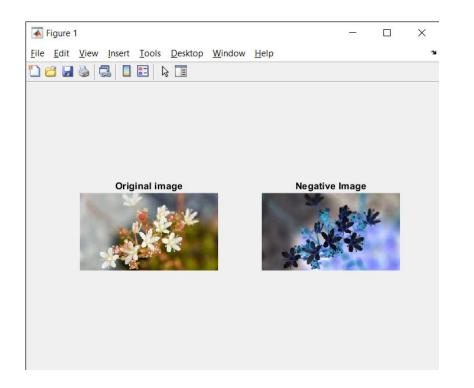


# AIM-2: Inspect the effect of image negative.

## **CODE:**

```
Editor - D:\Sem-6\CV\Practical-1\negativeImage.m
    colorToGrayImage.m \hspace{0.2in} \hspace{0.2in} \hspace{0.2in} \hspace{0.2in} negativeImage.m \hspace{0.2in} \hspace{0.2in} \hspace{0.2in} \hspace{0.2in} contrastStreching.m \hspace{0.2in} \hspace{0.2in} \hspace{0.2in} \hspace{0.2in} noiseImage.m
          % reading the RGB file into the Matlab environment
         skI = imread("colorImage.jpg");
 2 -
          subplot(1, 2, 1),
 3 -
 4
 5
         % displaying the RGB image
        imshow(skI);
         title("Original image");
 7 -
 8
 9
         % levels of the 8-bit image
          L = 2 ^ 8;
10 -
11
12
         % finding the negative
         neg = (L - 1) - skI;
13 -
14 -
         subplot(1, 2, 2),
15
16
         % displaying the negative image
17 -
        imshow(neg);
18 -
          title("Negative Image")
```

# **OUTPUT:**



# AIM-3: Perform contrast stretching and contrast threshold. Analyze the results for different values of (r1,s1) and (r2,s2).

#### **CODE:**

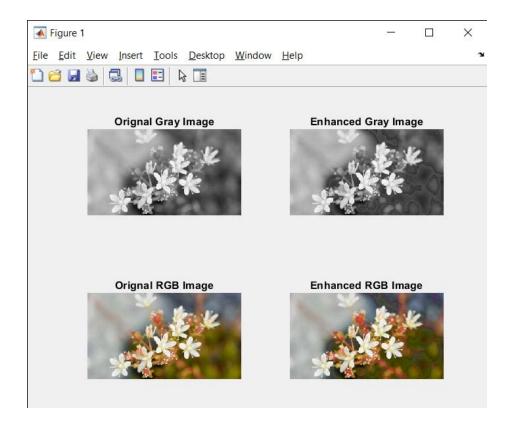
If r1 = 80 and w1 = 50:

```
Editor - D:\Sem-6\CV\Practical-1\contrastStreching.m
 +1 negativeImage.m × contrastStreching.m × noiseImage.m × SmoothingFilter.m
 1 -
       clear all;
 2 -
       close all:
 3 -
       clc
 4
 5
       % reading the RGB file into the Matlab environment
 6 -
      img = imread('colorImage.jpg');
 7 -
       grayImg = rgb2gray(img);
 8
 9
       % declaring variables for formula
10 -
       w1 = 50;
       w2 = 150;
11 -
12 -
       r1 = 80;
13 -
       r2 = 150;
14 -
       L = 255;
15
16 -
       a = w1/r1;
17 -
       b = (w2-w1)/(r2-r1);
18 -
       g = (L-w2)/(L-r2);
19
20 -
       [x, y, z] = size(qrayImq);
21
22
       % showing orignal Image
23 -
       subplot(2, 2, 1)
24 -
       imshow(gravImg);
25 -
       title('Orignal Gray Image');
26
```

```
Editor - D:\Sem-6\CV\Practical-1\contrastStreching.m
+1 negativeImage.m × contrastStreching.m × noiseImage.m × SmoothingFilter.m × intro
       title('Orignal Gray Image');
26
27
28 - ☐ for i = 1:x
29 - for j = 1:y
30 -
               if grayImg(i,j) <= r1</pre>
31 -
                   r = grayImg(i,j);
32 -
               elseif grayImg(i,j) > r1 && grayImg(i,j) <= r2</pre>
33 -
                   r = grayImg(i,j);
34 -
                   grayImg(i,j) = (b*(r-r1))+w1;
35 -
               else
36 -
                    r = grayImg(i,j);
37 -
                    grayImg(i,j) = (g*(r-r2))+w2;
38 -
               end
39 -
           end
      end
40 -
41
42
       %showing enhanced Image
43 -
       subplot(2 , 2, 2)
44 -
       imshow(grayImg);
45 -
       title('Enhanced Gray Image');
46
47
48 -
       subplot(2, 2, 3)
49 -
       imshow(img);
50 -
        title('Orignal RGB Image');
```

```
Editor - D:\Sem-6\CV\Practical-1\contrastStreching.m
    negativeImage.m × contrastStreching.m × noiseImage.m × SmoothingFilter.m × intro
44 -
        imshow(grayImg);
45 -
       title('Enhanced Gray Image');
46
47
48 -
      subplot(2, 2, 3)
49 -
      imshow(img);
50 -
      title('Orignal RGB Image');
51 - \Box \text{ for } k = 1:3
52 - =
53 - =
           for i = 1:x
               for j = 1:y
54 -
                    if img(i,j,k) <= r1</pre>
55 -
                        r = img(i,j,k);
56 -
                    elseif img(i,j,k) > r1 \&\& img(i,j,k) <= r2
57 -
                        r = img(i,j,k);
58 -
                         img(i,j,k) = (b*(r-r1))+w1;
59 -
                    else
60 -
                        r = img(i,j,k);
61 -
                         img(i,j,k) = (g*(r-r2))+w2;
62 -
                    end
63 -
                end
64 -
            end
65 -
66
67 -
        subplot(2, 2, 4)
68 -
        imshow(img);
69 -
        title('Enhanced RGB Image');
```

# **OUTPUT:**



If r1 = 100 and w1 = 20:

49 -

50 -

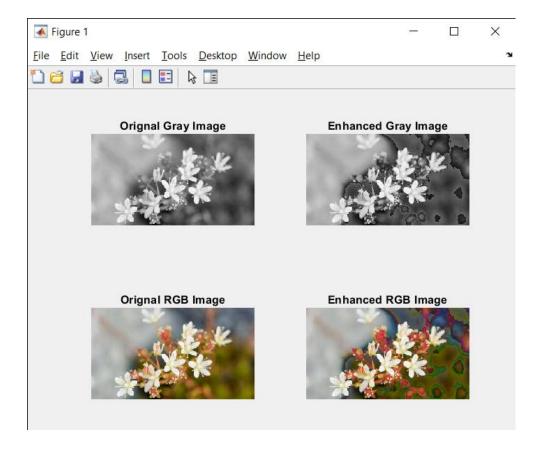
imshow(img);

title('Orignal RGB Image');

```
Editor - D:\Sem-6\CV\Practical-1\contrastStreching.m
                    × contrastStreching.m × noiseImage.m × SmoothingFilter.m × introd
+1 negativelmage.m
 2 -
        close all;
 3 -
       clc
 4
 5
       % reading the RGB file into the Matlab environment
 6 -
      img = imread('colorImage.jpg');
 7 -
       grayImg = rgb2gray(img);
 8
       % declaring variables for formula
10 -
       w1 = 20;
11 -
       w2 = 150;
       r1 = 100;
12 -
       r2 = 150;
13 -
       L = 255;
14 -
1.5
16 -
       a = w1/r1;
17 -
       b = (w2-w1)/(r2-r1);
18 -
       g = (L-w2)/(L-r2);
19
20 -
       [x, y, z] = size(grayImg);
21
22
       % showing orignal Image
23 -
       subplot(2, 2, 1)
24 -
       imshow(grayImg);
25 -
        title('Orignal Gray Image');
26
Editor - D:\Sem-6\CV\Practical-1\contrastStreching.m
+1 negativelmage.m × contrastStreching.m × noiselmage.m × SmoothingFilter.m × intro
25 -
       title('Orignal Gray Image');
26
27
28 - □ for i = 1:x
29 - for j = 1:y
30 -
              if grayImg(i,j) <= r1</pre>
31 -
                    r = grayImg(i,j);
32 -
               elseif grayImg(i,j) > r1 && grayImg(i,j) <= r2</pre>
33 -
                   r = grayImg(i,j);
34 -
                    grayImg(i,j) = (b*(r-r1))+w1;
35 -
36 -
                   r = grayImg(i,j);
37 -
                    grayImg(i,j) = (g*(r-r2))+w2;
38 -
                end
39 -
           end
40 -
41
42
       %showing enhanced Image
43 -
      subplot(2 , 2, 2)
44 -
       imshow(grayImg);
45 -
       title('Enhanced Gray Image');
46
47
48 -
       subplot(2, 2, 3)
```

```
Editor - D:\Sem-6\CV\Practical-1\contrastStreching.m
 +1 negativelmage.m × contrastStreching.m × noiselmage.m × SmoothingFilter.m ×
44 -
45 -
        imshow(grayImg);
       title('Enhanced Gray Image');
46
47
48 -
49 -
        subplot(2, 2, 3)
        imshow(img);
50 -
        title('Orignal RGB Image');
51 - \bigcirc \text{for } k = 1:3
52 - -
53 - -
54 -
            for i = 1:x
                 for j = 1:y
                     if img(i,j,k) <= r1</pre>
55 -
                          r = img(i,j,k);
56 -
57 -
58 -
                      elseif img(i,j,k) > r1 && img(i,j,k) <= r2
                          r = img(i,j,k);
                          img(i,j,k) = (b*(r-r1))+w1;
59 -
                      else
60 -
                          r = img(i,j,k);
61 -
62 -
                          img(i,j,k) = (g*(r-r2))+w2;
                      end
63 -
                 end
64 –
65 –
             end
       end
66
67 -
        subplot(2, 2, 4)
68 -
       imshow(img);
69 -
       title('Enhanced RGB Image');
```

### **OUTPUT:**



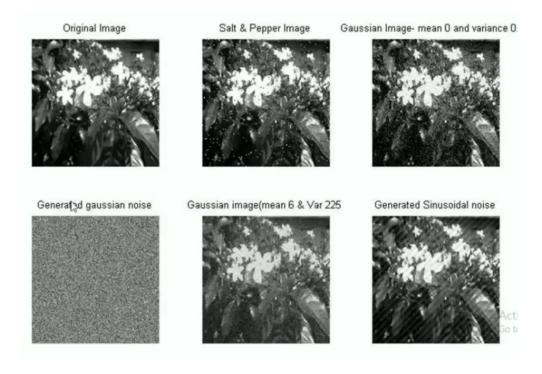
# AIM-4: Add noise to your own image.

#### **CODE:**

```
clc
1
          close all
2
          clear all
3
4
          % Read the test Image
5
          img = imread('colorImage.jpg');
6
7
          mygrayimg = rgb2gray(img);
          mygrayimg = imresize(mygrayimg,[256 256], 'nearest');
8
9
          subplot(2,3,1);
          imshow(mygrayimg);
10
          title('Original Image');
11
12
13
          % Add Salt and pepper noise with noise density 0.02
          salt = imnoise(mygrayimg, 'salt & pepper', 0.02);
14
          subplot(2,3,2);
15
          imshow(salt);
16
17
          title('Salt & Pepper Image');
18
          % Add Gaussian noise with mean 0 and variance 0.01
19
          gau = imnoise(mygrayimg, 'gaussian', 0, 0.01);
20
21
          subplot(2,3,3);
22
          imshow(gau);
23
          title('Gaussian Image- mean 0 and variance 0.01');
```

```
tmonow(Paa)
23
          title('Gaussian Image- mean 0 and variance 0.01');
24
          % Generate Gaussian noise with mean 6 and variance 225
25
          mynoise = 6 + sqrt(225) * randn(256,256);
26
27
          subplot(2,3,4);
28
          imshow(mynoise,[]);
          title('Generated gaussian noise');
29
30
31
          % Original Image and generated Gaussian
32
          subplot(2,3,5);
33
          mynoiseimg = double(mygrayimg) + mynoise;
34
35
          imshow(mynoiseimg,[]);
          title('Gaussian image(mean 6 & Var 225');
36
37
          % Original Image plus sinusoidal noise
38
39
40
          subplot(2,3,6);
          [x y] = meshgrid(1:256,1:256);
41
          mysinusoidalnoise = 15 * \sin(2*pi/14*x+2*pi/14*y);
42
43
          mynoiseimg1 = double(mygrayimg) + mysinusoidalnoise;
          imshow(mynoiseimg1,[]);
44
45
          title('Generated Sinusoidal noise');
```

## **OUTPUT:**



AIM-5: Implement different spatial domain smoothening filters with different mask size. Analyze the result.

## **CODE:**

If We apply 7, 7 mask size:

```
clear all;
 1
 2
          close all;
 3
          clc;
 4
 5
          img = imread('colorImage.jpg');
          grayImg = rgb2gray(img);
 6
 7
          w = zeros(7, 7);
 8
 9
10
          for i = 1:7
              for j = 1:7
11
12
                  w(i,j) = 1/70;
                  outputImg = imfilter(img, w);
13
              end
14
15
          end
16
17
          subplot(2, 2, 1)
18
          imshow(img)
          title('Orignal Image');
19
20
21
          subplot(2, 2, 2)
22
          imshow(outputImg)
23
          title('Smoothing Image');
```

## **OUTPUT:**

Orignal Image



Smoothing Image



If we apply 5, 5 mask size:

```
clear all;
1
          close all;
 2
 3
          clc;
4
 5
          img = imread('colorImage.jpg');
6
          grayImg = rgb2gray(img);
7
8
          w = zeros(5, 5);
9
10
          for i = 1:5
              for j = 1:5
11
12
                  w(i,j) = 1/70;
                  outputImg = imfilter(img, w);
13
14
              end
15
          end
16
          subplot(2, 2, 1)
17
          imshow(img)
18
19
          title('Orignal Image');
20
          subplot(2, 2, 2)
21
          imshow(outputImg)
22
          title('Smoothing Image');
23
```

# **OUTPUT:**







# **LEARNING OUTCOME:**

In this practical I learned about how we can convert image into gray scale, image into negative, image starching, image thresholding, image noises and image smoothing.