



Department of Computer Science and Engineering (Data Science)
Image Processing and Computer Vision I (DJ19DSL603)

Lab 3: Image Enhancement in Spatial Domain using Point Processing Techniques

Name: Yash Thakar

SAP ID.: 60009210205

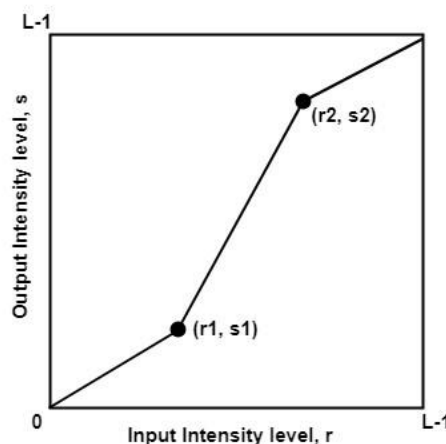
Aim: To perform image enhancement in spatial domain using point processing techniques: contrast stretching, log transformation, power law transformation

Theory:

1. Contrast Stretching:

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 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.

Below figure shows a typical transformation function used for Contrast Stretching.



By changing the location of points (r_1, s_1) and (r_2, s_2) , we can control the shape of the transformation function. For example,

- When $r_1 = s_1$ and $r_2 = s_2$, transformation becomes a **Linear function**.



Department of Computer Science and Engineering (Data Science)

- ii. When $r_1=r_2$, $s_1=0$ and $s_2=L-1$, transformation becomes a **thresholding function**.
- iii. When $(r_1, s_1) = (r_{\min}, 0)$ and $(r_2, s_2) = (r_{\max}, L-1)$, this is known as **Min-Max Stretching**.
- iv. When $(r_1, s_1) = (r_{\min} + c, 0)$ and $(r_2, s_2) = (r_{\max} - c, L-1)$, this is known as **Percentile Stretching**.

Image Processing and Computer Vision I (DJ19DSL603)

Lab 3: Image Enhancement in Spatial Domain using Point Processing Techniques

2. Log Transformation:

The log transformations can be defined by this formula:

$$s = c \log(r + 1)$$

Where s and r are the pixel values of the output and the input image and c is a constant.

The value 1 is added to each of the pixel value of the input image because if there is a pixel intensity of 0 in the image, then $\log(0)$ is equal to infinity. So 1 is added, to make the minimum value at least 1.

During log transformation, the dark pixels in an image are expanded as compare to the higher pixel values. The higher pixel values are kind of compressed in log transformation.

3. Power Law Transformation:

The power law transformation, also called gamma transformation is a technique that uses a power-law function to adjust the pixel values of an image. It is versatile, as it allows the emphasis on certain intensity ranges or enhancing specific details in an image.

Formula of the gamma transformation equation is given as:



Department of Computer Science and Engineering (Data Science)

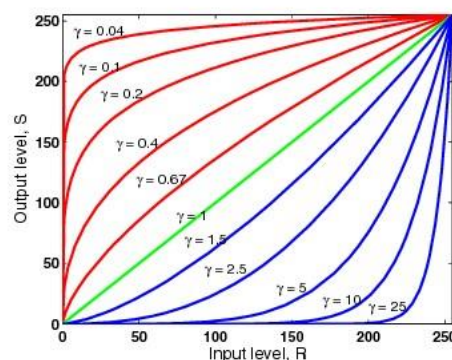
$$O = c * I^\gamma$$

In this context, the symbol c is used to represent the scaling factor, γ denotes the gamma correction value, and I and O respectively stand for the input and output pixel values.

If the value of gamma is greater than one, it stretches contrast in brighter areas and compresses the pixel values in the darker areas. If the value of gamma is smaller than one, it enhances contrast in dim areas and compresses the pixel values in the bright areas. The graphical representation of the function is given below:

Image Processing and Computer Vision I (DJ19DSL603)

Lab 3: Image Enhancement in Spatial Domain using Point Processing Techniques



Here, the x-axis and y-axis represent the input and output pixel values respectively and the shape of the graph varies with the gamma values.

Lab Assignments to complete in this session

Problem Statement: Develop a Python program utilizing the OpenCV library to enhance the images in spatial domain. The program should address the following tasks:

1. Read any low contrast image of your choice or generate one.
2. Display the before & after image(s) used in every task below.
3. Apply contrast stretching.
4. Apply Log transformation.



Shri Vile Parle Kelavani Mandal's

DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING

(Autonomous College Affiliated to the University of Mumbai)

NAAC Accredited with "A" Grade (CGPA : 3.18)



Department of Computer Science and Engineering (Data Science)

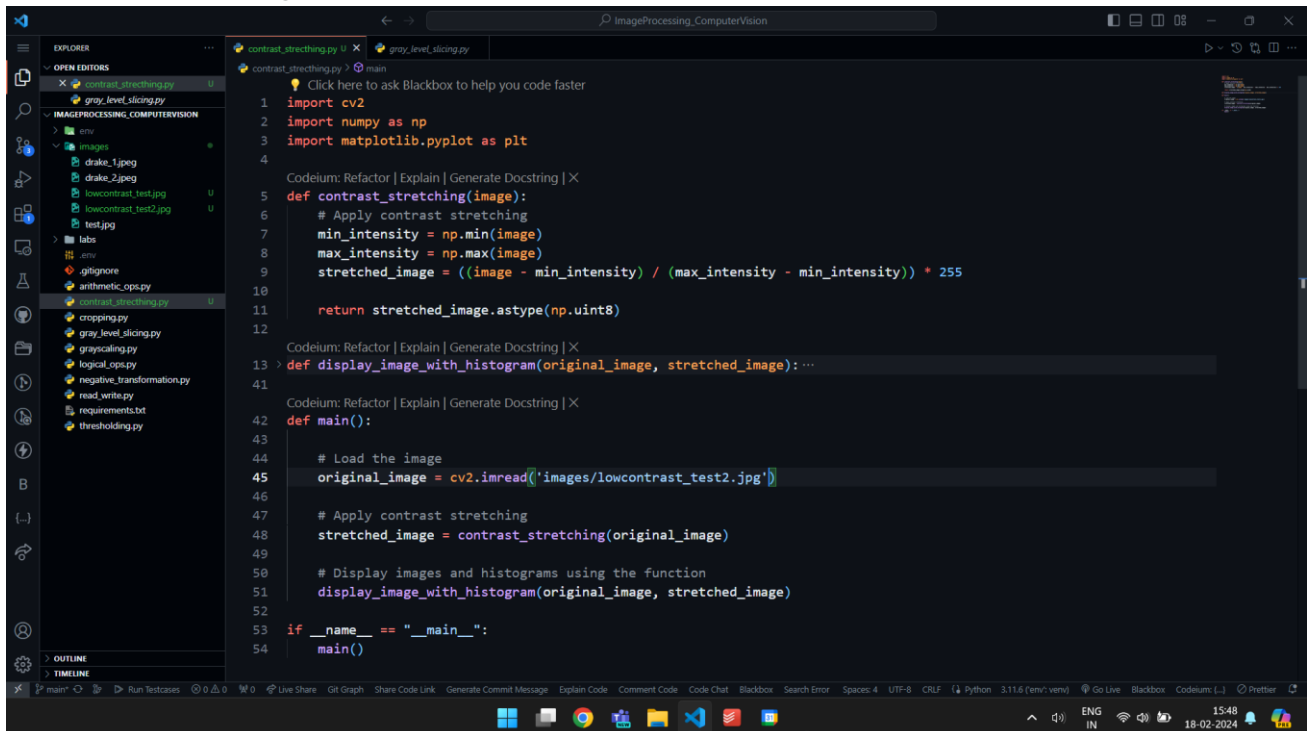
5. Apply Power law transformation
6. Perform all the task for black and white images and on coloured RGB Images on all the three channels.

The solution to the operations performed must be produced by scratch coding without the use of built in OpenCV methods.



Department of Computer Science and Engineering (Data Science)

Contrast Stretching:

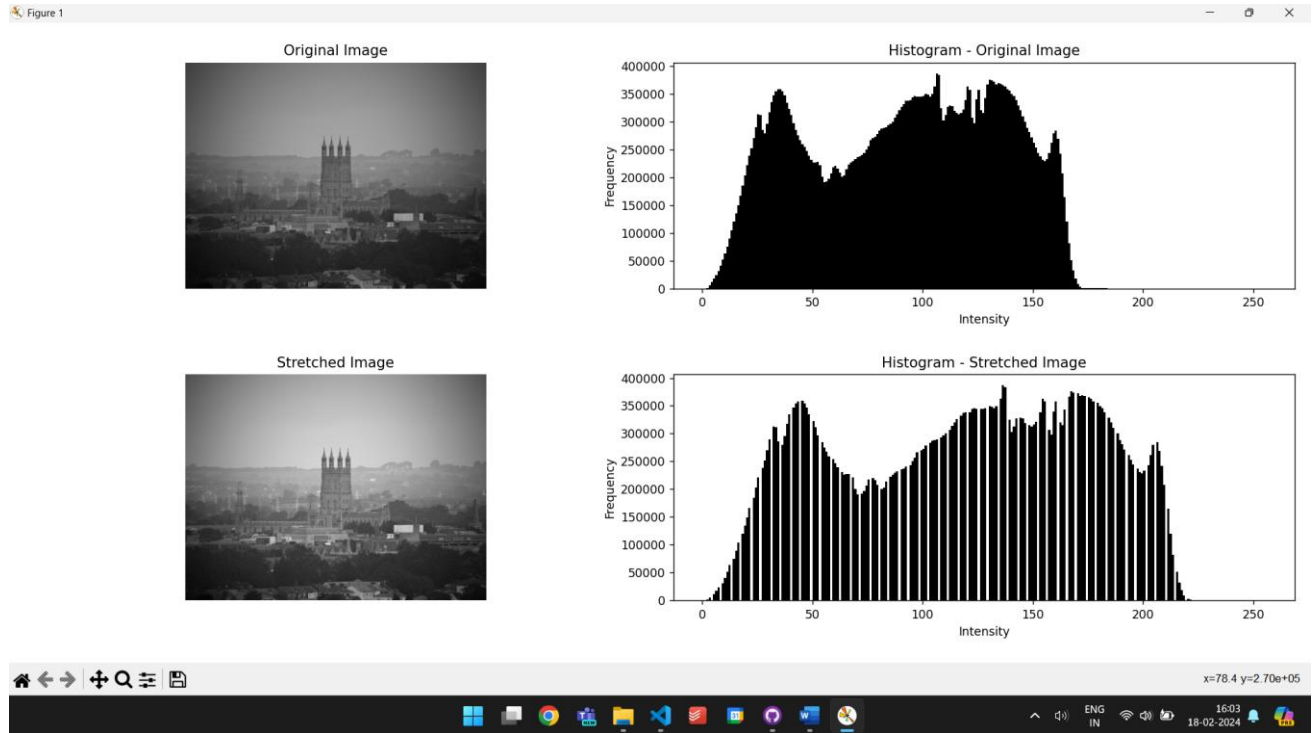


```
1 import cv2
2 import numpy as np
3 import matplotlib.pyplot as plt
4
5 def contrast_stretching(image):
6     # Apply contrast stretching
7     min_intensity = np.min(image)
8     max_intensity = np.max(image)
9     stretched_image = ((image - min_intensity) / (max_intensity - min_intensity)) * 255
10
11     return stretched_image.astype(np.uint8)
12
13 def display_image_with_histogram(original_image, stretched_image):...
14
15 def main():
16     # Load the image
17     original_image = cv2.imread('images/lowcontrast_test2.jpg')
18
19     # Apply contrast stretching
20     stretched_image = contrast_stretching(original_image)
21
22     # Display images and histograms using the function
23     display_image_with_histogram(original_image, stretched_image)
24
25 if __name__ == "__main__":
26     main()
```

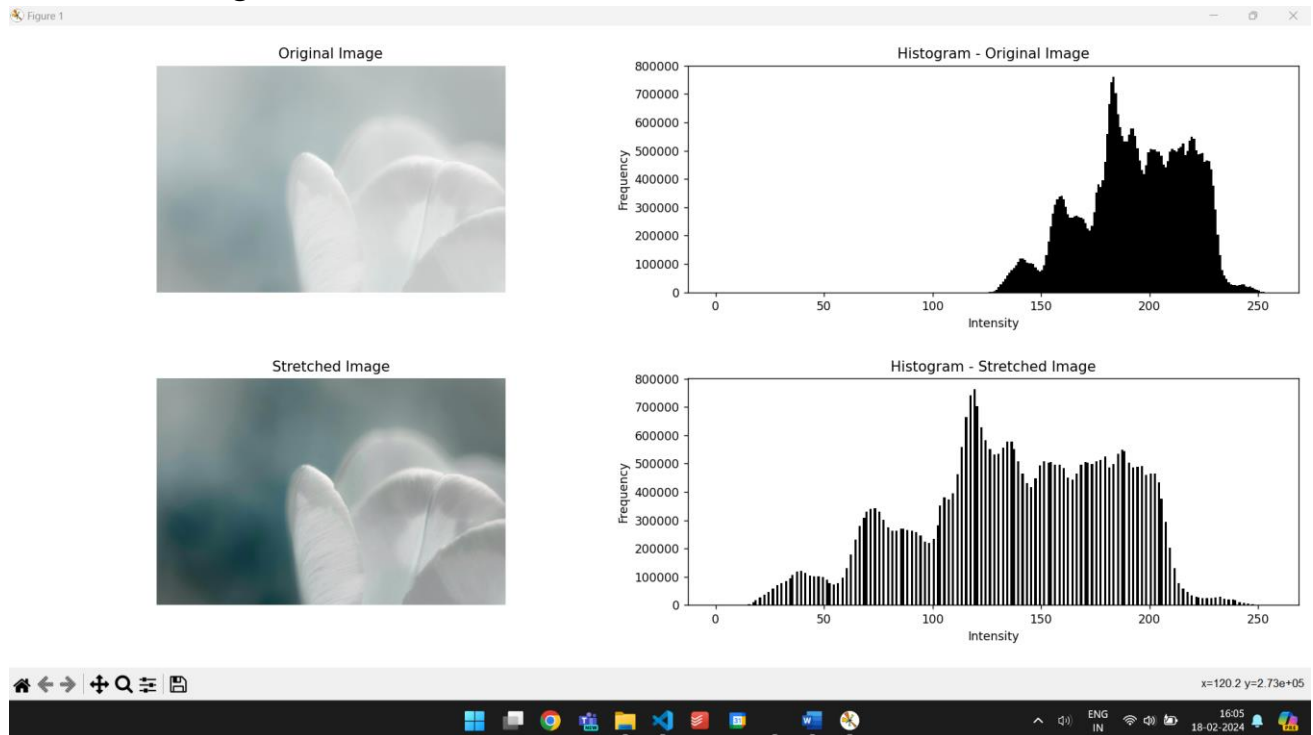


Department of Computer Science and Engineering (Data Science)

For Grayscale image:



For colour image:





Department of Computer Science and Engineering (Data Science)

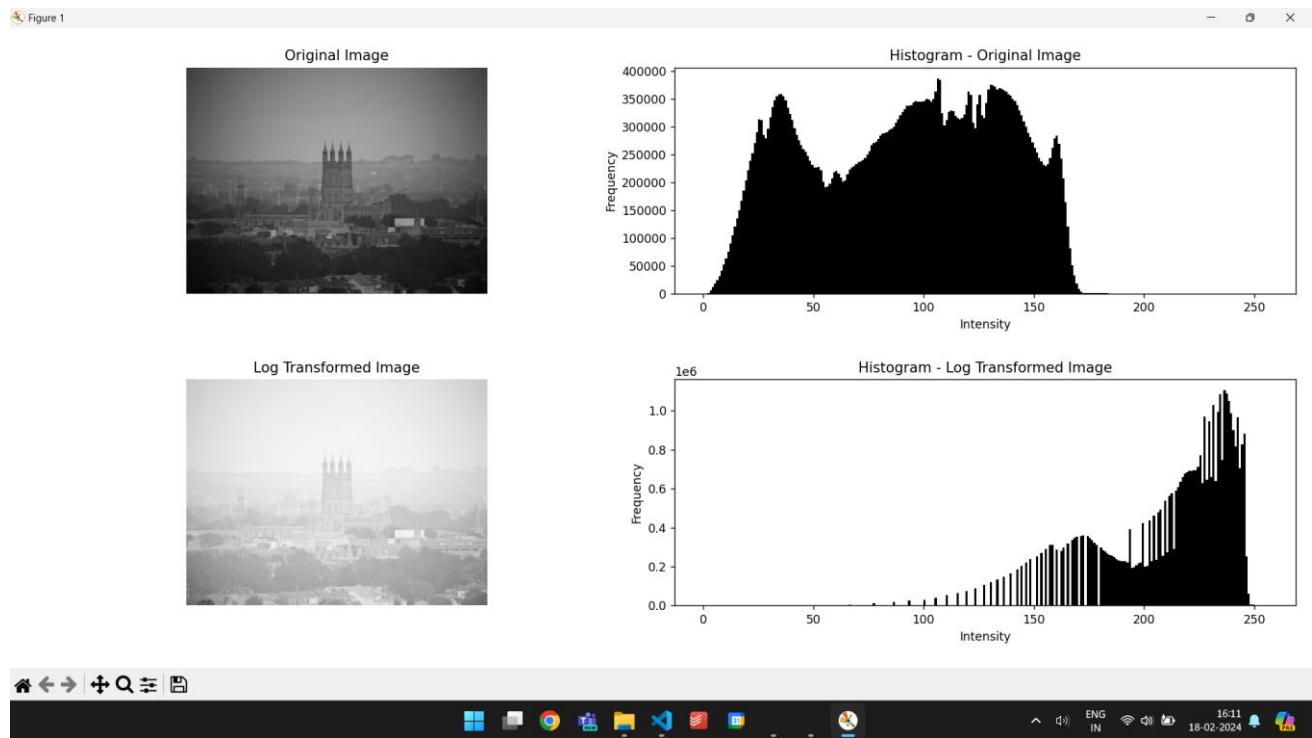
Log Transformation

```
1 import cv2
2 import numpy as np
3 import matplotlib.pyplot as plt
4
5 def log_transformation(image):
6     # Apply log transformation
7     log_image = np.log1p(image)
8
9     # Normalize to 0-255 range
10    log_image = (log_image / np.max(log_image)) * 255
11
12    return log_image.astype(np.uint8)
13
14 def display_image_with_histogram(original_image, transformed_image):
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43 def main():
44     # Load the image
45     original_image = cv2.imread('images/lowcontrast_test.jpg')
46
47     # Apply log transformation
48     log_transformed_image = log_transformation(original_image)
49
50     # Display images and histograms using the function
51     display_image_with_histogram(original_image, log_transformed_image)
52
53 if __name__ == "__main__":
54     main()
55
```

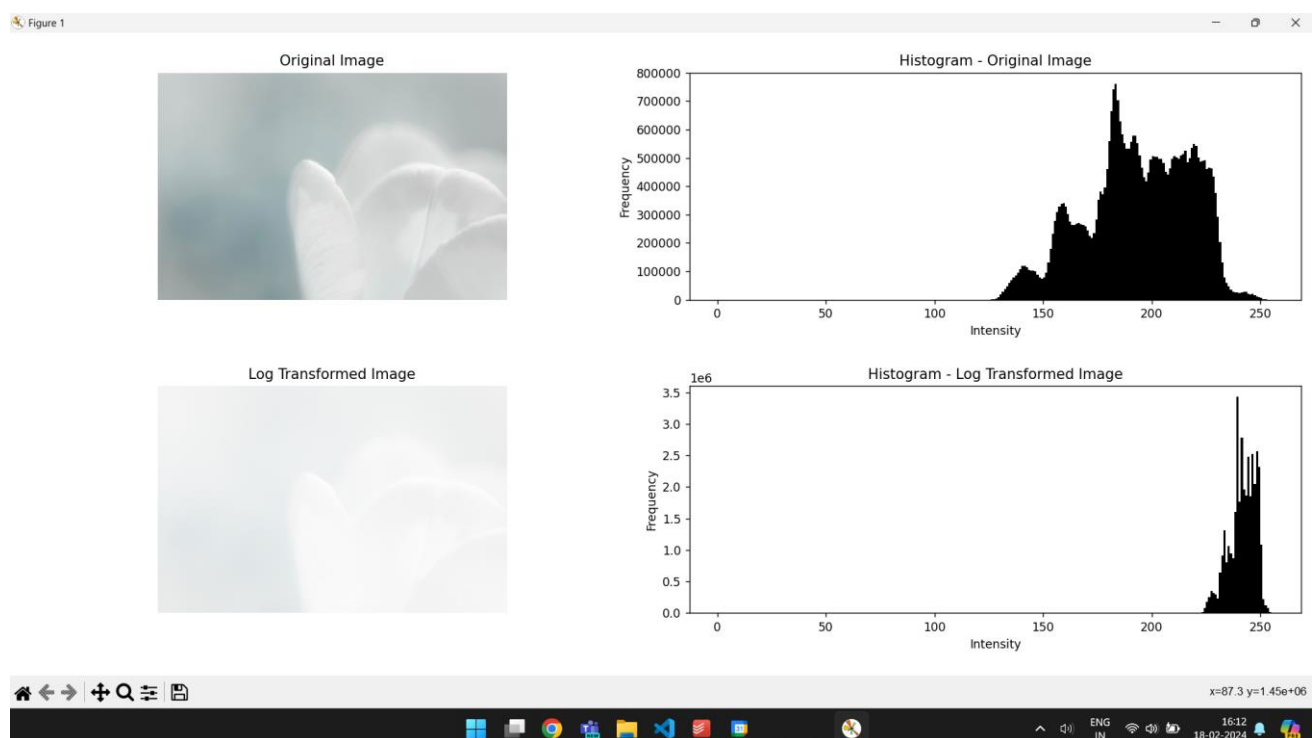


Department of Computer Science and Engineering (Data Science)

For Grayscale images:



For colour images:





Department of Computer Science and Engineering (Data Science)

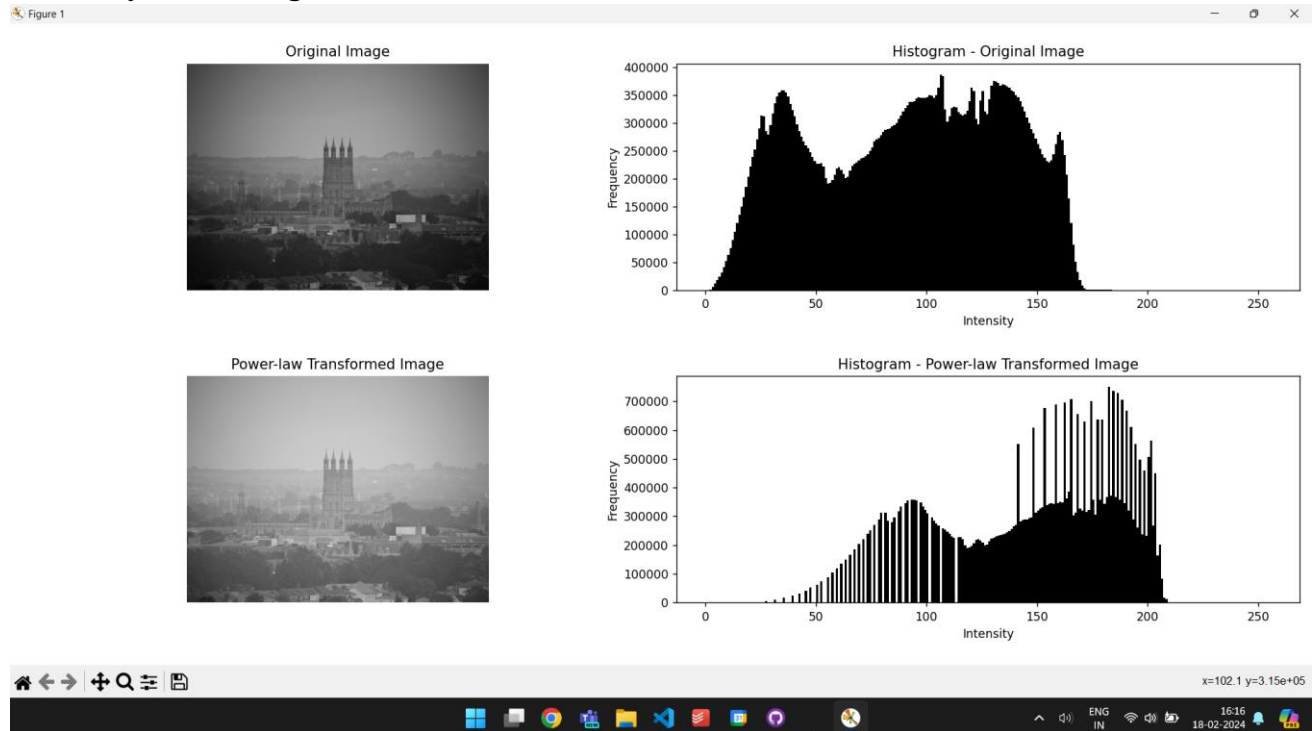
Power law transformation

```
1 import cv2
2 import numpy as np
3 import matplotlib.pyplot as plt
4
5 def power_law_transformation(image, gamma):
6     power_law_image = np.power(image / 255.0, gamma) * 255.0
7
8     return power_law_image.astype(np.uint8)
9
10 def display_image_with_histogram(original_image, transformed_image):
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29 def main():
30     original_image = cv2.imread('images/lowcontrast_test2.jpg', cv2.IMREAD_GRAYSCALE)
31
32
33     # Define the gamma value for power-law transformation
34     gamma = 0.5
35     power_law_transformed_image = power_law_transformation(original_image, gamma)
36
37     # Display images and histograms using the function
38     display_image_with_histogram(original_image, power_law_transformed_image)
39
40
41 if __name__ == "__main__":
42     main()
43
44
45
46
47
48
49
50
51
```



Department of Computer Science and Engineering (Data Science)

For Grayscale image:



For colour image:

