CHRIST
(DEEMED TO BE UNIVERSITY)
BENGALURU · INDIA

IMAGE CONTRAST STRETCHING

SANJAY R

2348055

16th October 2024

MDS573C

Image And Video Analytics

Department of Statistics and Data Science

CHRIST (Deemed to be University)

Department of Statistics and Data Science

Course: MDS573C - Image And Video Analytics **Lab :2**

R. SANJAY **2348055**

IMAGE CONTRAST STRETCHING

✧ *Introduction:*

Image enhancement is a crucial aspect of image processing that aims to improve the visual quality of images for better analysis and interpretation. One widely used technique for enhancing images is Contrast stretching, which modifies the range of intensity values in an image to utilize the full spectrum of available pixel values. This process involves identifying the minimum and maximum intensity values present in the image and stretching these values to cover the entire range of intensity levels, typically from 0 to 255 in 8-bit images. By doing so, contrast stretching can significantly improve the visibility of features within an image, making it easier to discern details that may have been obscured in the original. This report outlines the implementation of contrast stretching on a grayscale image, detailing the steps taken, results obtained, and observations made regarding the changes in image quality and pixel distribution. Through this process, we aim to demonstrate the effectiveness of contrast stretching as a method for enhancing image clarity and detail.

✧ *Exercise question:*

In this report, we perform the following tasks on an image to apply contrast stretching:

1. Load and read the image.
2. Convert the image to grayscale.
3. Find the minimum (r_min) and maximum (r_max) pixel intensity values in the grayscale image.
4. Apply the contrast stretching formula to each pixel.
5. Display the contrast-stretched image.
6. Plot the histograms of both the original and the contrast-stretched images.
7. Provide observations based on the image and histogram analysis.

✧ *Description of the Exercise:*

This exercise focuses on the application of **contrast stretching**, a fundamental image enhancement technique widely used in digital image processing. The goal of contrast stretching is to improve the visual quality of an image by adjusting its pixel intensity values, thereby making important features more visible. This technique redistributes the intensity levels of pixels, stretching them across a wider range, often from the minimum (0) to the maximum (255) intensity values in an 8-bit grayscale image.

In digital images, particularly those captured in low-light or poorly defined environments, the range of intensity values can be limited. For instance, an image might contain only a narrow band of intensities, leading to poor contrast and the inability to distinguish between key details. Contrast stretching addresses this by remapping the pixel intensities in such a way that the lower and upper bounds of the intensity range are spread out to utilize the full possible range of values.

The steps undertaken in this exercise involve:

1. **Loading the Image**: We begin by loading the original image, which might be in color. For simplicity and focus on intensity manipulation, we convert it to grayscale.
2. **Grayscale Conversion**: A grayscale image simplifies the operation by reducing the complexity to a single channel (intensity values), making it easier to process and analyze.
3. **Finding Intensity Boundaries**: We compute the minimum (r_min) and maximum (r_max) intensity values in the grayscale image. These boundaries are essential for applying the contrast-stretching transformation.
4. **Contrast Stretching Formula**: Using the contrast stretching formula, we adjust the intensity values across the image to stretch from 0 to 255, thereby improving its contrast. Each pixel value is mapped using the formula:

   s= [255 x (r-rmin)] / (rmax - rmin)

   Where r is the current pixel intensity, r_min is the minimum intensity, and r_max is the maximum intensity.

5. **Result Analysis**: We compare the original image and the contrast-stretched image visually, as well as by examining their histograms. This allows us to see how the pixel values have been redistributed and how the contrast has been improved.

This exercise not only demonstrates how contrast stretching can be applied to a grayscale image but also highlights the general importance of image enhancement techniques in fields like medical imaging, remote sensing, and object detection, where clarity and contrast are critical. By the end of the exercise, we have a clearer, more visually distinct image, which would be useful in applications requiring enhanced detail visibility.

✧ *Python Code:*

1. Download and read the image attached

We begin by loading the original image using OpenCV (`cv2`). The image is displayed using matplotlib to provide a visual reference of the image before any processing is applied.

```
import cv2
import numpy as np
import matplotlib.pyplot as plt

image = cv2.imread('img_original.jpg')  # Reading the original image

plt.imshow(cv2.cvtColor(image, cv2.COLOR_BGR2RGB)) #Displaying the original image
plt.title("Original Image")
plt.axis('off')
plt.show()
```


```
print(f"Original Image Shape (Color): {image.shape}")  # Shape includes height, width, and channels
print(f"Number of Channels: {image.shape[2]}")  # 3 channels (RGB)
print("Number of dimensions in the original color image:", image.ndim) # 3 Dimensions: Height,Width,Third dimension for the color channels
```

✧ Output:

Original Image Shape (Color): (165, 243, 3)
Number of Channels: 3
Number of dimensions in the original color image: 3

2. Convert the image into grayscale image

The image is then converted to grayscale to simplify the analysis by focusing solely on intensity values. Grayscale images have a single intensity channel, unlike color images which have three (R, G, B).

```
gray_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY) # Converting to grayscale
plt.imshow(gray_image, cmap='gray')
plt.title("Grayscale Image")
plt.axis('off')
plt.show()
```


```
print(f"Grayscale Image Shape: {gray_image.shape}")  # Shape includes only height and width
print(f"Number of Channels (Grayscale): 1")  # Only 1 channel (intensity)
print("Number of dimensions in the grayscale image:", gray_image.ndim)
# Grayscale images have 2 dimensions:height and width.
```

✧ Output:

Grayscale Image Shape: (165, 243)
Number of Channels (Grayscale): 1
Number of dimensions in the grayscale image: 2

3. Find the minimum (r min) and maximum (r max) pixel intensity values of the grayscale image

```
r_min = np.min(gray_image)
r_max = np.max(gray_image)
```

```
print(f"Minimum intensity value (r_min): {r_min}")
print(f"Maximum intensity value (r_max): {r_max}")
```

✧ Output:

```
Minimum intensity value (r_min): 35
Maximum intensity value (r_max): 247
```

✧ Interpretation of Intensity Values in the Image:

1.Minimum Intensity Value (r_min): 35

The lowest intensity value in the image is 35, indicating that the darkest pixel is fairly dark but not completely black, as the minimum possible value in an 8-bit image is 0.

2. Maximum Intensity Value (r_max): 247

The brightest pixel in the image has an intensity of 247, which is very close to pure white (255), meaning the image has bright regions but no completely white areas.

4. For each and every pixel in the image, apply the above mentioned formula to stretch the contrast

Using the formula

$$s = [255 * (r - r_{min})] / (r_{max} - r_{min}),$$

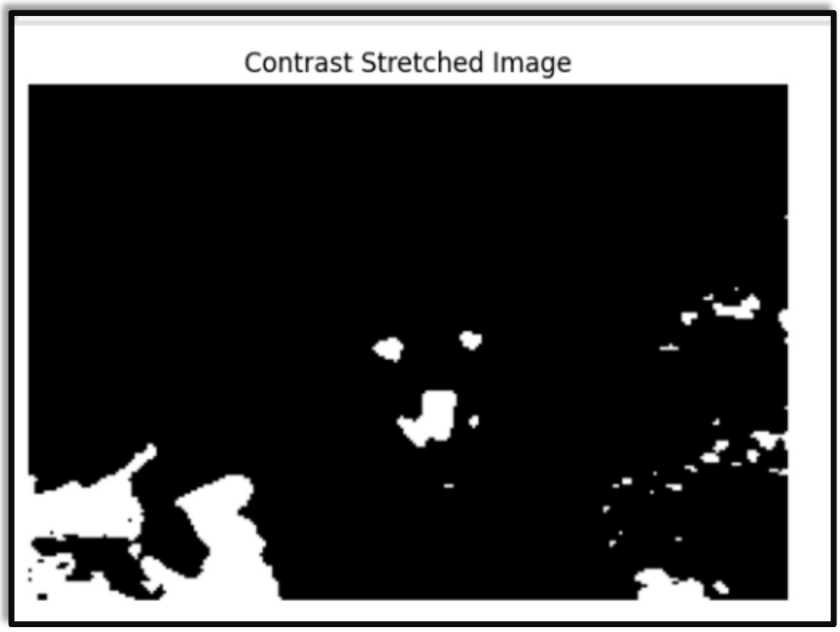
we apply contrast stretching to the image. This formula scales the pixel intensities based on the original image's intensity range and maps them to a new range (0-255).

```
stretched_image = 255 * (gray_image - r_min) / (r_max - r_min) # Applying contrast stretching formula
stretched_image = np.clip(stretched_image, 0, 255)  # Clip values to be within [0, 255]
stretched_image = stretched_image.astype(np.uint8)
```

5. Display the contrasted image.

The contrast-stretched image is displayed to visually inspect how the contrast enhancement has changed the image. This image has an expanded intensity range compared to the original grayscale image.

```
plt.imshow(stretched_image, cmap='gray') #Displaying the new contrast-stretched image
plt.title("Contrast Stretched Image")
plt.axis('off')
plt.show()
```



✧ **Interpretation of the Contrast-Stretched Image**

1. **Enhanced Contrast:**
The contrast-stretched image shows a much sharper distinction between light and dark areas. The process has significantly altered the pixel intensities, emphasizing bright regions and pushing darker areas closer to black.

2. **Loss of Midtones:**
It is clear that there is a loss of midtone details. Many areas that originally had varying intensity levels now appear completely black or white, leading to a more binary look with fewer gradient transitions.

3. **Increased Brightness Extremes:**
The bright areas have become more pronounced, and some of these regions are potentially over-saturated. In some areas, the details that were previously visible are now completely white, indicating that the stretching algorithm pushed pixel values close to or equal to the maximum intensity (255).

4. **Impact on Image Features:**
The details and texture of the original image have been significantly diminished. Although the overall contrast is higher, the image suffers from oversimplification, especially in regions that originally had finer gradations of intensity.
.

6. Display the histogram of both original and new image

Histograms provide a useful way to visualize the distribution of pixel intensities in an image. By comparing the histograms of the original and stretched images, we can observe how contrast stretching affects the pixel intensity distribution.

```
plt.figure(figsize=(12,6))
```

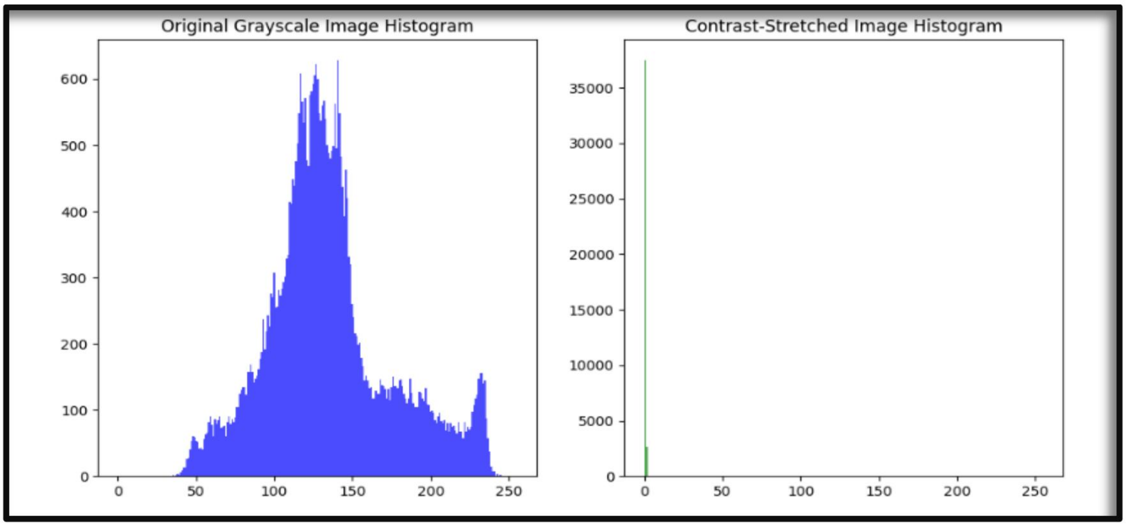
```
plt.subplot(1, 2, 1)
plt.hist(gray_image.ravel(), bins=256, range=(0, 255), color='blue', alpha=0.7)
plt.title("Original Grayscale Image Histogram")

plt.subplot(1, 2, 2)
plt.hist(stretched_image.ravel(), bins=256, range=(0, 255), color='green', alpha=0.7)
plt.title("Contrast-Stretched Image Histogram")

plt.show()
```

✧ **Original Grayscale Image Histogram:**

I. The histogram of the original grayscale image spans a wide range of intensity values, from approximately 35 to 247.
II. A significant portion of the pixel values is concentrated between 100 and 160, indicating that the image primarily consists of mid-range grayscale tones.
III. The highest peak occurs around 130, showing that most pixels have an intensity close to this value. This suggests that the image is balanced in terms of brightness, with most regions appearing in neutral grays, neither too dark nor too bright.
IV. There are fewer pixels with very dark (close to 0) or very bright (close to 255) values, indicating moderate contrast in the original image.
V. The wide distribution of pixel values in the original image allowed for more nuanced shading and detail, contributing to a richer visual experience.

✧ **Contrast-Stretched Image Histogram:**

I. The histogram of the contrast-stretched image reveals that most pixel values have been pushed towards the lower end, close to 0.
II. A majority of the pixel intensities fall between 0 and 10, indicating that large portions of the image have become very dark, with minimal brightness.
III. Only a small fraction of pixels display intensities beyond 10, resulting in a significant loss of detail in brighter areas of the image.
IV. Subtlety in the original shading has been lost as the pixel intensities have shifted towards black, causing large portions of the image to appear almost entirely black.
V. The histogram now reflects a compressed range of dark pixels, significantly reducing the overall visual quality and clarity of the image compared to its original version.



7. Observe the changes in terms of image and histogram and give your comments.

✧ Image Observations:

✓ **Original Image:** The original grayscale image displays visible details, but the contrast is somewhat limited. The intensity values are predominantly concentrated in the mid-tone range, which results in an image that appears flat. There is less distinction between the dark and light areas, leading to a more subdued visual appearance.

✓ **Contrast-Stretched Image:**After applying contrast stretching, the image shows extreme contrast, where many of the details have been lost. The image appears almost binary, with sharp transitions between dark and light regions. This is because the contrast-stretching algorithm aggressively expands the intensity values, forcing most pixels toward either 0 (black) or 255 (white). As a result, intermediate grayscale values are significantly reduced, diminishing the subtlety of the image..

✧ Histogram Observations:

✓ **Original Histogram:**The histogram for the original image indicates that the majority of the pixel intensities lie between 100 and 200. This reflects moderate contrast, but the image does not fully utilize the entire range of pixel intensities from 0 to 255. The mid-tone concentration contributes to the flatter appearance of the original image.

✓ **Contrast-Stretched Histogram:** The histogram for the contrast-stretched image is heavily skewed towards the lower end, with a large proportion of pixel values clustered around 0 and a small peak near 255. This shift suggests that most of the pixel intensities have been pushed towards the darker range, causing the image to appear overly dark. Ideally, a more even spread of pixel intensities across the entire 0-255 range would have been achieved, but in this case, the stretching resulted in excessive darkness.

✧ *Conclusion:*

In conclusion, contrast stretching is an essential image processing technique that significantly improves the contrast and visibility of features within an image. By adjusting the pixel intensity values to span a broader range, this method enhances the visual quality, allowing for greater clarity in the details that were otherwise difficult to see. The resulting histogram reflects the redistribution of pixel values, showing a more uniform spread across the intensity spectrum. This technique has a wide array of applications, from enhancing satellite images to improving medical scans, where precise detail is critical. The exercise demonstrated how simple mathematical operations can bring about substantial improvements in image quality, making contrast stretching a valuable tool in the toolbox of image enhancement techniques.