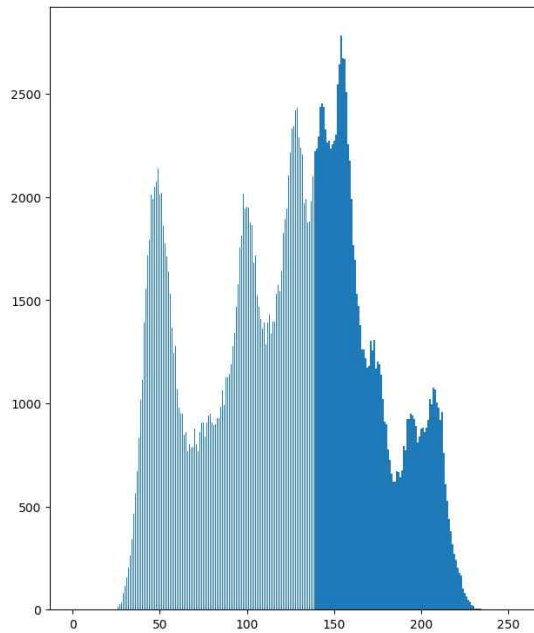


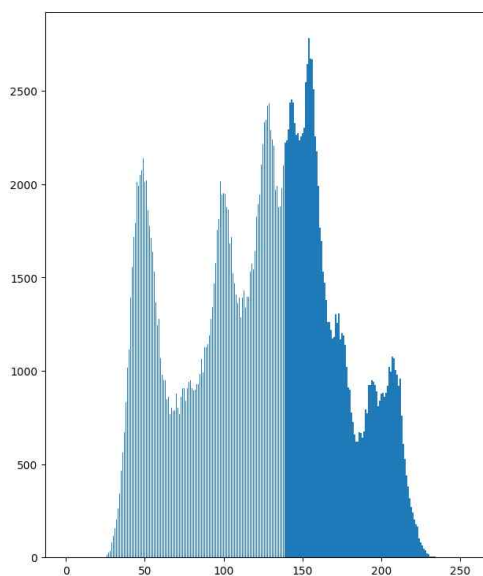
영상처리 hw#2 보고서

From all the above results, discuss differences among Histogram Equalization, Basic Contrast Stretching, and Ends-in Contrast Stretching.

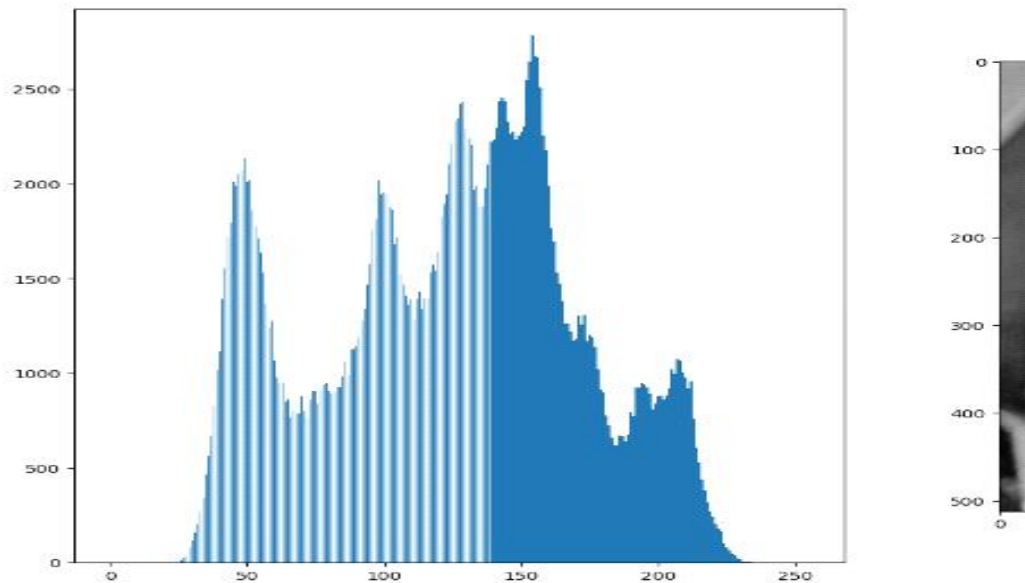
(a) Plot the histogram.



(b) Perform Histogram Equalization and then plot the histogram.



```
13 draw_graph(raw_data, input_file, output_path, True)
```



```
def hist_equalization(raw_data, count_num_list):
    LUT = [0 for i in range(len(count_num_list))]

    accumulate_sum = 0
    for i in range(len(count_num_list)):
        # 히스토그램 누적 합 구하기
        accumulate_sum += count_num_list[i]

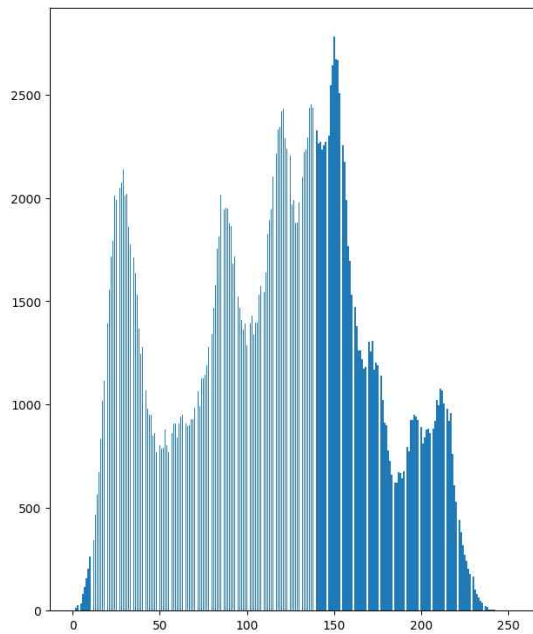
        # 정규화를 통해 패핑에 사용할 LUT 생성
        LUT[i] = int(255 * accumulate_sum / (512 * 512) + 0.5)

    if LUT[i] > 255:
        LUT[i] = 255
    elif LUT[i] < 0:
        LUT[i] = 0

    # LUT를 이용하여 결과 값 생성
    hist_data = [0 for i in range(len(raw_data))]
    for i in range(len(raw_data)):
        hist_data[i] = LUT[raw_data[i]]

    return hist_data
```

- (c) Perform Basic Contrast Stretching and then plot the histogram.
Discuss any difference when compared to the result of (b).



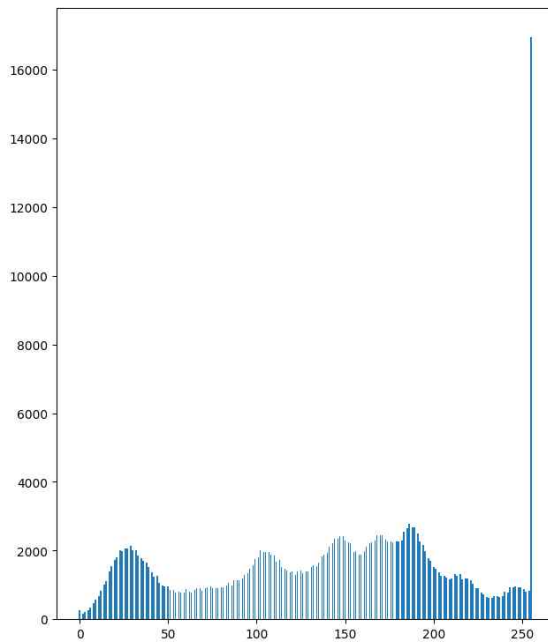
```
#기존의 이미지 대비 스트레칭을 하는 함수
def basic_contrast_stretching(raw_data):
    hist_data = [0 for i in range(len(raw_data))]
    min_val = 256
    max_val = -1

    for i in range(len(raw_data)):
        if min_val > raw_data[i]:
            min_val = raw_data[i]
        if max_val < raw_data[i]:
            max_val = raw_data[i]

    for i in range(len(raw_data)):
        hist_data[i] = int(255 * (raw_data[i] - min_val) / (max_val - min_val) + 0.5)
        if hist_data[i] > 255:
            hist_data[i] = 255
        elif hist_data[i] < 0:
            hist_data[i] = 0

    return hist_data
```

(d) Perform Ends-in Contrast Stretching with low-end value of 30 and high-end value of 200, and then plot the histogram. Discuss any difference when compared to the result of (c).



```
[34] 1 #끝에서 대비 스트레칭 함수
      2 def endsin_contrast_stretching(raw_data, min_val, max_val):
      3     hist_data = [0 for i in range(len(raw_data))]
      4
      5     for i in range(len(raw_data)):
      6         hist_data[i] = int(255 * (raw_data[i] - min_val) / (max_val - min_val) + 0.5)
      7         if hist_data[i] > 255:
      8             hist_data[i] = 255
      9         elif hist_data[i] < 0:
      10             hist_data[i] = 0
      11
      12     return hist_data
      13
```

```
1 #이미지 출력 및 저장
2 input_file = '/content/drive/MyDrive/영상처리과제/lena_bmp_512x512_new.bmp'
3 output_graph_path = '/content/drive/My Drive/영상처리과제/HW#2_Output_D_Plot_Histogram'
4 new_img_path = '/content/drive/My Drive/영상처리과제/Endsin_Contrast_Stretching_lena_D.bmp'
5
6 BMPHEADERS, raw_data = get_bmp_data(input_file)
7 #끝에서 대비 스트레칭 값은 low-end value of 30 and high-end value of 200 이다.
8 min_val = 30
9 max_val = 200
10 hist_data = endsin_contrast_stretching(raw_data, min_val, max_val)
11
12 create_bmp_img(BMPHEADERS, hist_data, new_img_path)
13 draw_graph(hist_data, new_img_path, output_graph_path, True)
```

결과 비교 및 보고

레나 사진을 히스토그램으로 표현하는 과제이였습니다. A는 단순히 레나 사진을 히스토그램으로 표현한 것 입니다. B는 A를 바탕으로 평활화(Equalization)하는 과정

이였습니다 다만 사진 결과에서 봐듯이 멀리서 보면 A와 큰 차이가 없으며, 확대해서 봐야 B는 평활화 된 것을 알 수 있습니다. C는 기본 대비 스트레칭 분석이며 그 결과를 B와 비교하는 문제입니다. 기본 대비 스트레칭은 이미지의 픽셀 값을 선형적으로 늘려서 사용 가능한 전체 동적 범위까지를 커버하는 방법이며, 이미지 내 최소값과 최대값을 기준으로 픽셀 값을 선형적으로 확장하여 밝기 수준의 범위를 확장 합니다. 이러한 방법은 간단하고 구현하고 쉽지만, 이미지내의 픽셀 값이 분포가 고르지 않은 경우 항상 최적의 결과를 낼 수 없습니다. 그리고 C와 B를 결과를 비교해보면 C가 B보다 조금 더 고르게 퍼진 히스토그램을 확인할 수 있습니다. D는 끝에서 대비 스트레칭 문제이며 히스토그램의 극단 부분만 스트레칭하되, 중간 값은 변경하지 않은 방법입니다. 전체 히스토그램을 스트레칭하는 것이 아니라 최소 또는 최대 강도만 조정하여 전체 값 범위를 확장합니다. 이러한 방법은 이미지 내 대부분의 픽셀 값이 잘 분포 되어 있지만, 일부 구역에 낮거나 높은 강도가 있는 경우에 유용합니다. C와는 비교해보면 C는 고르게 분포했으나, D는 최소 값이나, 최대값에서 값이 크게 하여 히스토그램을 구현했다는 것을 알 수 있습니다. D는 최소 값이 30, 최대 값이 200으로 지정해줬기 때문에, 값을 설정해서 `def endsin_contrast_stretching()` 함수에 값을 전달하여 뽑아 줍니다.

이러한 분석을 통해 히스토그램에서 평활화를 하면 픽셀 강도를 재부분배하고, 더욱 균일한 히스토그램을 얻을려고 합니다. 그리고 기본 대비 스트레칭은 전체 픽셀 값 범위를 선형적으로 늘려주고, 반대로 끝에서 대비 스트레칭은 픽셀 값 범위의 극단 부분만 조정하기 때문에 각각의 방법은 서로 다른 장점이 있으며, 상황에 따라 적합한 방법으로 적용하면 된다는 것을 알 수 있습니다.

<참조>

1.

<https://ko.wikipedia.org/wiki/%ED%9E%88%EC%8A%A4%ED%86%A0%EA%B7%B8%EB%9E%A8> - 위키피디아 히스토그램

2.

영상처리 강의자료 2장

3.

<https://kr.mathworks.com/help/images/histogram-equalization.html> - MathWorks 문서(매트랩) 히스토그램 평활화를 사용하여 영상대비 조정하기