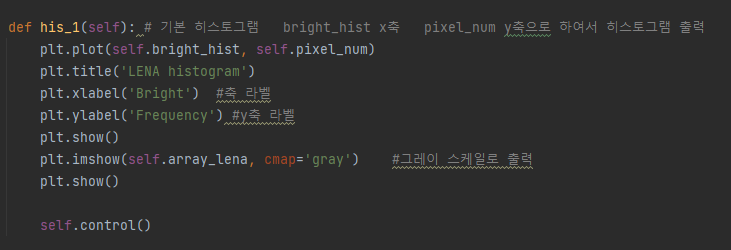
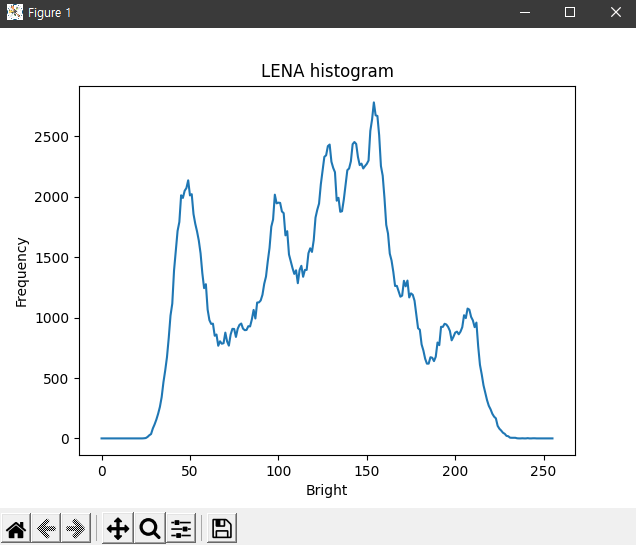
**영상처리 HW #2**

2018253096 류승현

**1. Plot the histogram**

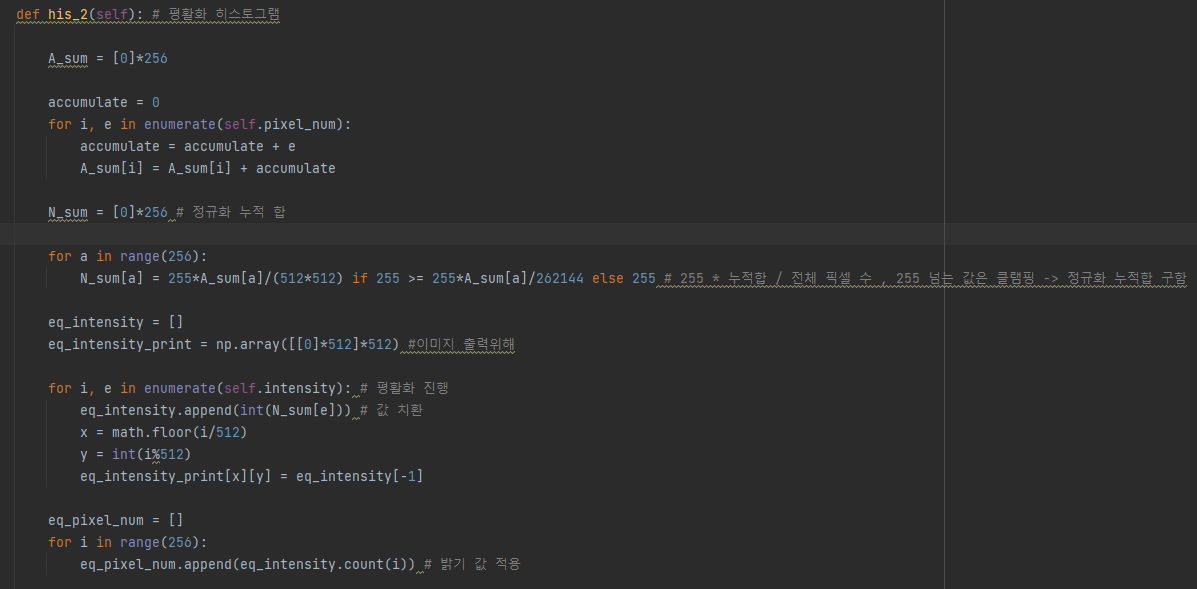


* 메뉴에서 기본 히스토그램 출력 메뉴를 선택하면 위의 코드가 실행이됨
* 기본적인 세팅 부분은 마지막에 서술하였음.
* x축에 들어갈 정보와 y축에 들어갈 정보를 받아와서 plt.plot을 이용해서 출력해주었음, imshow()를 통해 이미지 출력

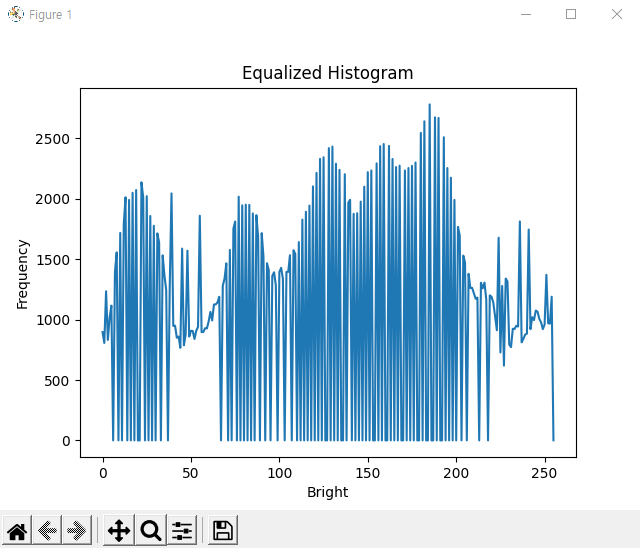


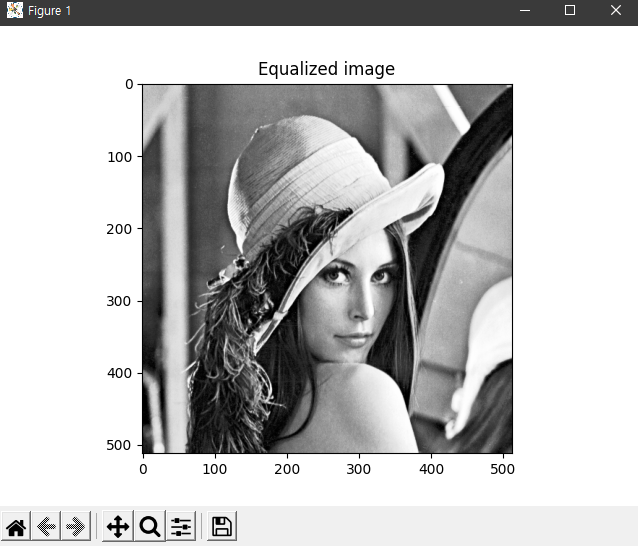
* 레나 이미지에 대한 히스토그램이 출력 된 것을 확인할 수 있었음.

**2.** Perform Histogram Equalization and then plot the histogram.



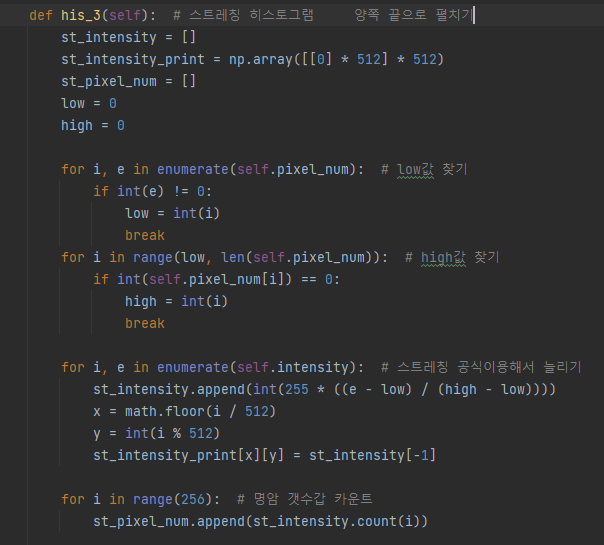
* 누적 합 값을 계산하여 A\_sum배열에 저장함.
* 정규화 된 누적 합을 구하기 위해서 N\_sum배열에 255 \* 누적합 / 픽셀수를 해주어서 값을 구해주고 255가 넘는다면 클램핑을 해주었음.
* 이미지 값 출력을 위해서 2차원 배열에 값 넣어줌
* 이후에 1번처럼 plot을 이용해서 출력 처리, imshow()를 통해 이미지 출력



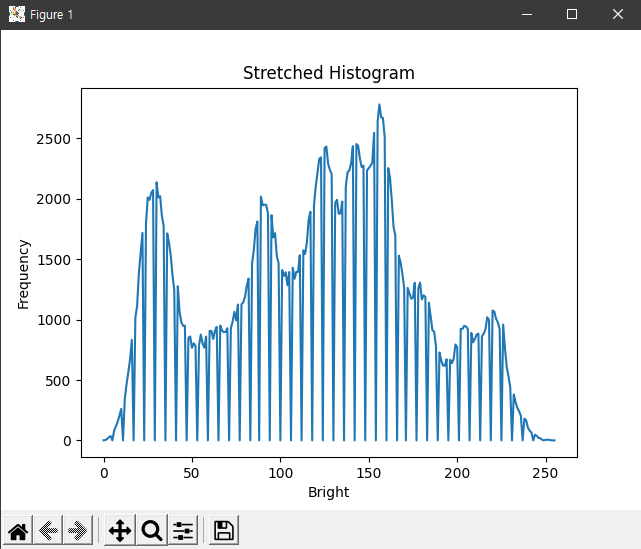
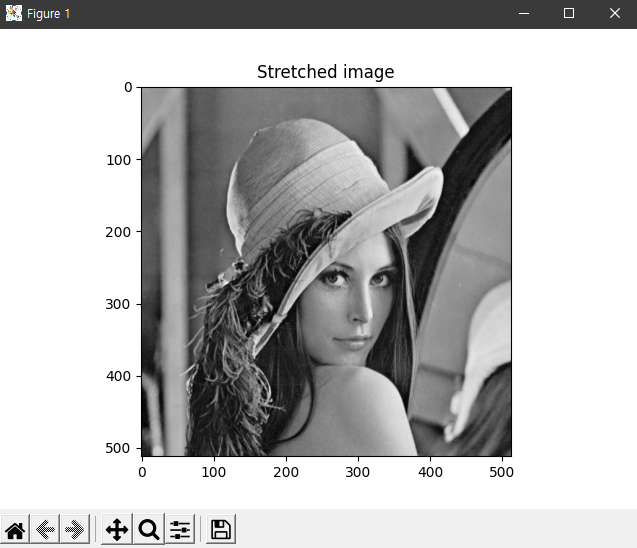


* 좀 더 균일하고 넓은 분포로 바뀐 것을 확인 할 수 있었음.

**3. Perform Basic Contrast Stretching and then plot the histogram.**

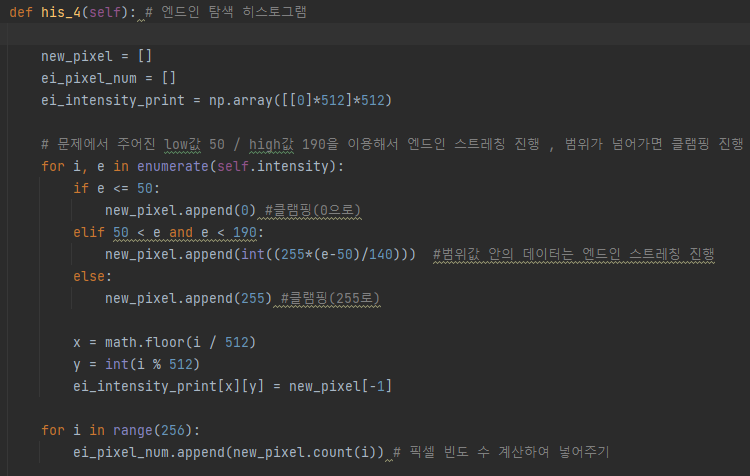
****

* for문을 이용해서 low / high값을 각각의 변수에 찾아서 저장해줌
* 수업시간에 배웠던 스트레칭 공식을 이용해서 스트레칭한 픽셀을 리스트에 담고, 각 명암 값 별로 픽셀 수를 세서 1번과 같이 plot을 이용해 출력해주었음, imshow()를 통해 이미지 출력

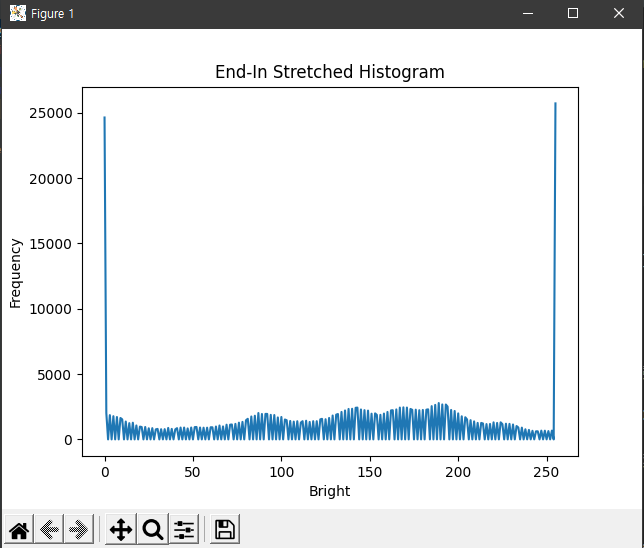
 

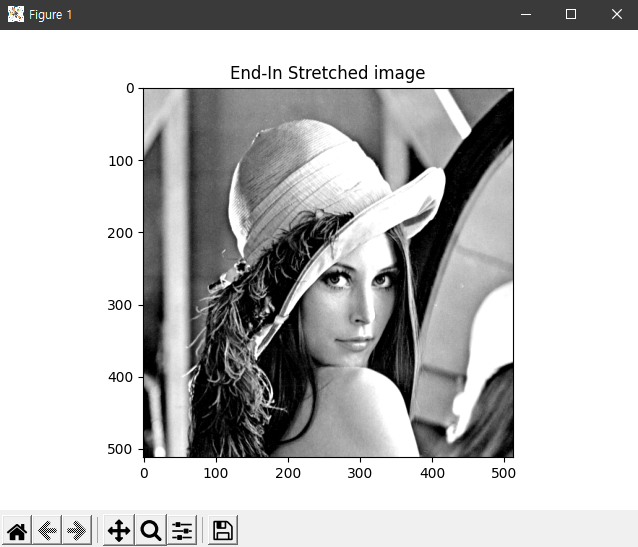
* 히스토그램을 보면 양 옆으로 늘렸기 때문에 0~255까지 값을 고루 가지는 것을 볼 수 있었음.
* 위의 2번 사진과 비교해보니 전체적으로 조금 어두워진 것 같았다. 골고루 밝기 값이 분포되어있는 느낌을 받을 수 있었다.

**4.** Perform Ends-in Contrast Stretching with low-end value of 50 and high-end value of 190, and then plot the histogram

****

* 문제에서 주어진 50~190을 활용해서 계산을 해주었음.
* if문을 이용해서 50이하면 0으로 / 190이상이면 255로 클램핑을 해주었음.
* 범위 안의 값이면 공식을 이용해서 스트레칭을 해주었음.
* 스트레칭 한 값으로 픽셀 수 빈도 수를 계산하여 다시 plot을 이용해서 그려줌, imshow()를 통해 이미지 출력

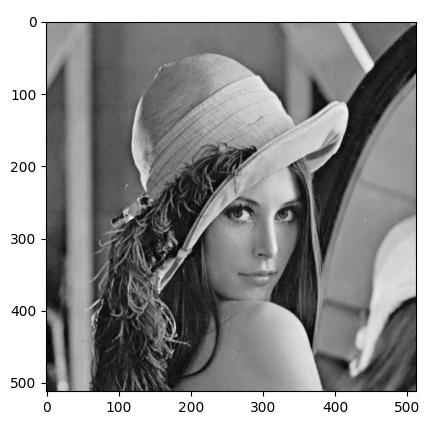




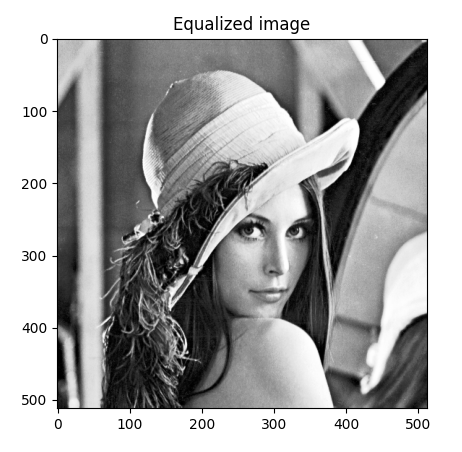
* 50~190 사이의 값을 쭉 늘려주고 범위를 벗어나는 값은 클램핑을 해주었기 때문에 0, 255값이 1자로 쭉 뻗어있는 모습의 히스토그램을 볼 수 있었다.

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

1)원본이미지



2) Equalization



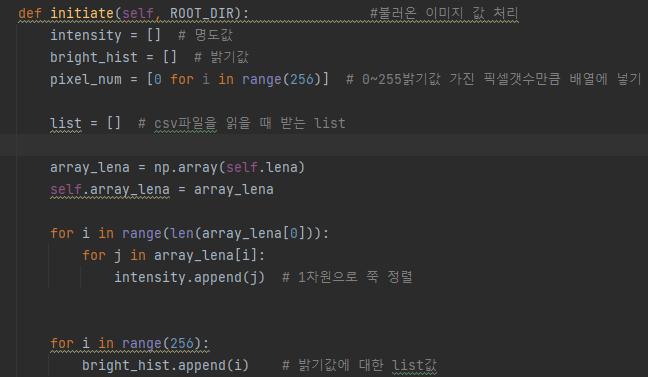
3) Basic Contrast Stretching



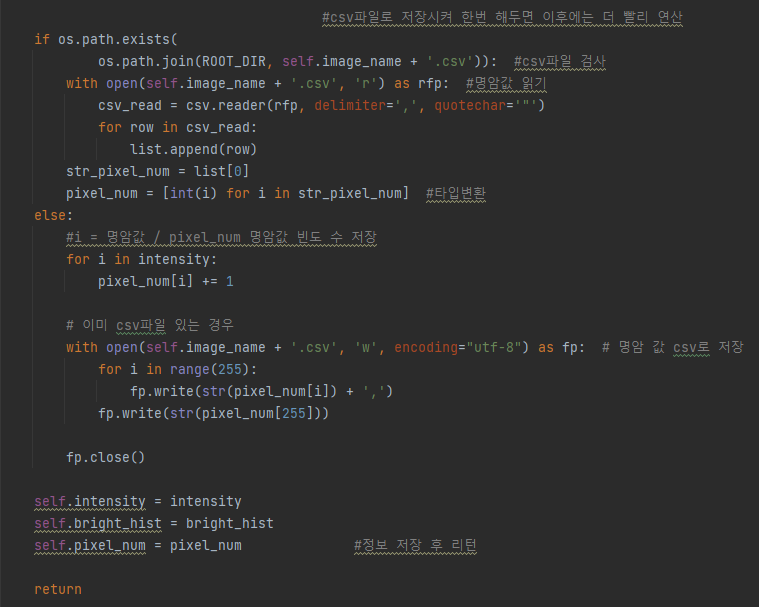
4) End –In Contrast Stretching



* 각 결과 이미지 값들을 확인해 보니 먼저 엔드인은 클램핑 된 값이 0,255들이 있어서 그런지 원본 이미지나 다른 이미지들에 비해서 밝은 부분은 더 밝고 어두운 부분은 더 어두웠다.
* 스트레칭 한 이미지는 너무 어둡거나 너무 밝거나 하지 않은 평균적인 밝기 값을 나타냈다.
* 평활화 이미지는 골고루 밝고 어두운 부분이 존재하였다.
* 불러온 이미지 값 처리에 대한 설명



* 이미지를 불러와서 픽셀 개수만큼 배열에 넣어주고 1차원으로 쭉 정렬시킴.



* 계속 같은 이미지를 처리하면 오래 걸리므로 처음에 csv로 만들어 놓고 불러와서 사용하는 방식
* x축에 들어갈 명암 값 / y축에 들어갈 빈도 수를 계산해서 리턴해줌.
* 각각의 영상 처리 시 이 값들을 불러와서 사용함.

참고자료 <https://kxmjhwn.tistory.com/248> , 강의자료