

영상처리 실습 보고서

3주차: Histogram equalization

학번	201802170
이름	하 상 호

1. 과제의 내용

5x5 행렬과 fruits.jpg 의 히스토그램 평활화

Histogram equalization 과정

1. histogram을 구한다.
 2. histogram을 총 픽셀수로 나눈다.
 3. 나눈값을 누적시킨다.
 4. 누적시킨 값에 gray level을 곱한다.
 5. 반올림 혹은 버림을 하여 정수값으로 변환한다.
 6. 구해진 정수값을 사용하여 histogram equalization 결과를 return
- 해당 과정의 함수 구현

my_calcHist(src)

my_normalize_hist(hist, pixel_num)

my_PDF2CDF(pdf)

my_denormalize(normalized, gray_level)

my_calcHist_equalization(denormalized, hist)

my_equal_img(src, output_gray_level)

2. 과제의 해결 방법

과제의 내용을 해결하기 위해 어떠한 방법을 사용했는지, 자세하게 기술한다.

1. histogram을 구한다.

image의 height width 값을 받고, image 의 weight를 히스토그램의 맞게 삽입
따라서 for 문을 사용하여 해당 pixel 에 접근하고 weight 값을 통해서 hist index에 1
씩 더해가며 히스토그램을 형성한다.

```
def my_calcHist(img):  
    (h, w) = img.shape  
    hist = np.zeros((256,))  
  
    for row in range(h):  
        for col in range(w):  
            hist[img[row][col]] += 1  
  
    return hist
```

2. histogram을 총 픽셀수로 나눈다.

normalized_hist 변수값에 각각의 hist 값을 총 픽셀 수로 나누어 배열을 형성
따라서 np.array 함수를 사용하여 hist 값을 하나하나 접근하여 총 픽셀 수로 나눈
값을 형성해준다.

```
def my_normalize_hist(hist, pixel_num):  
    normalized_hist = np.array([i/pixel_num for i in hist])  
    return normalized_hist
```

3. 나눈값을 누적시킨다.

pdf를 cdf로 변환 pdf와 같은 크기의 cdf 배열을 생성 한뒤 cdf 는 pdf의 index 0
에 위치한 값을 갖게 한다.

for 문을 통해서 현재 값은 이전값 + 현재값을 통해 나눈값을 누적시킨다.

```
def my_PDF2CDF(pdf):  
    cdf = np.zeros((256,))  
    cdf[0] = pdf[0]  
    for i in range(1, 256):  
        cdf[i] = pdf[i] + cdf[i-1]  
    return cdf
```

4. 누적시킨 값에 gray level을 곱한다.

normalize_hist 와 같은 방법이지만 나눗셈이 아니라 곱셈을 시행한다.

denormalized 변수를 np.array 함수를 통해 누적시킨값에 gray level을 곱하며
배열을 형성한다.

```
def my_denormalize(normalized, gray_level):  
    denormalized = np.array([i * gray_level for i in normalized])  
    return denormalized
```

5. 반올림 혹은 버림을 하여 정수값으로 변환한다.

본 과제에서는 구현이 되었지만 int 형으로 바꾸며 버림을 실행한다.

```
output_gray_level = denormalized_output.astype(int)
```

6. 구해진 정수값을 사용하여 histogram equalization 결과를 return
평활화 한 histogram을 return 해야 한다.

버림을 한 결과 값들을 통해서 같은 값들을 찾고 hist_equal 이라는 변수를 생성하여
해당 index 에 넣어주면된다.

실습 pdf 의 말을 빌리자면

ex)

hist[1] = 3

floor(denormalized[1]) = 1

hist[2] = 2

floor(denormalized[2]) = 1

floor(denormalized[1])과 floor(denormalized[2])가

서로 1로 같으므로 둘의 hist값을 합함

hist_equal[1] = hist[1] + hist[2] = 3 + 2 = 5

이를 for 문을 통해 해당 인덱스를 접근하며 더해준다.

```
def my_calcHist_equalization(denormalized, hist):  
  
    hist_equal = np.zeros((256,))  
    for i in range(256):  
        hist_equal[denormalized[i]] += hist[i]  
    return hist_equal
```

+) . 평활화 한 histogram을 통해 img를 출력

히스토그램을 만들었던 것처럼 h , w 값을 img를 통해 받는다.

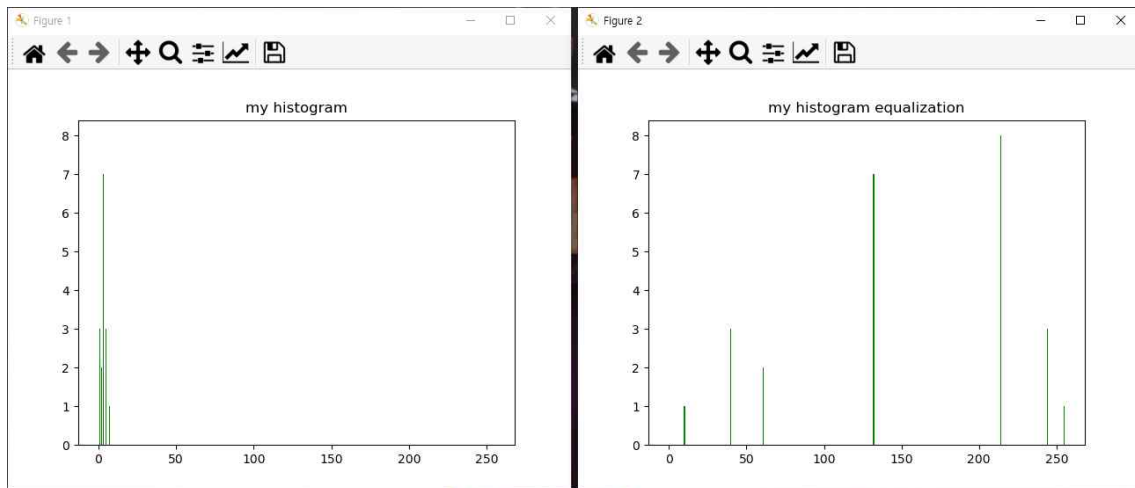
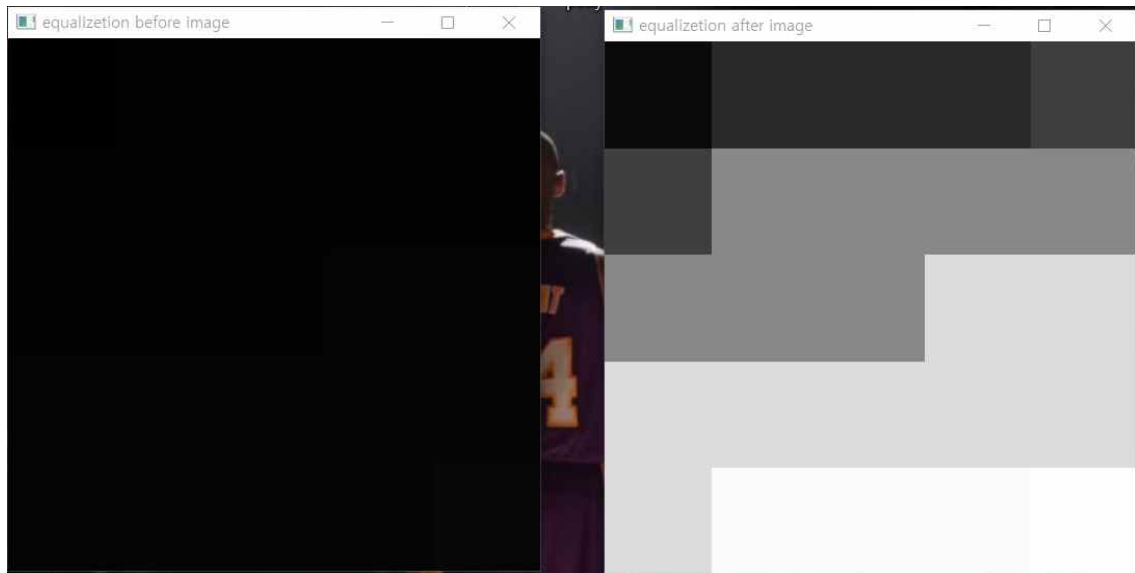
2중 for 문을 통해 새로운 이미지를 만들어 삽입한다.

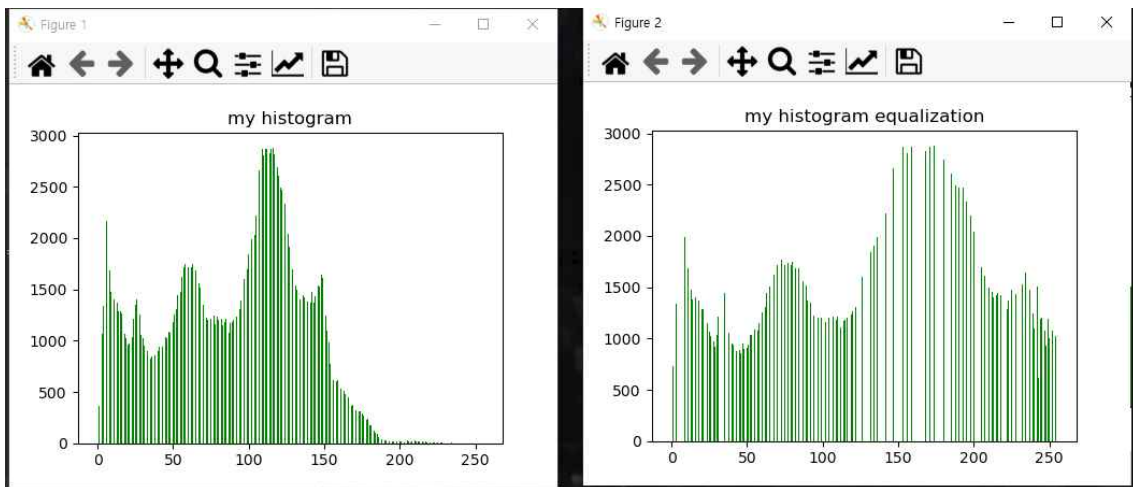
img 의 pixel 위치를 이용하여 gray_level 배열을 통해 해당 값들을 삽입

```
def my_equal_img(src, gray_level):  
    (h, w) = src.shape  
    dst = np.zeros((h, w), dtype=np.uint8)  
    for row in range(h):  
        for col in range(w):  
            dst[row, col] = gray_level[src[row, col]]  
    return dst
```

3. 결과물

결과물이 잘 보이도록 화면을 캡처해 보고서에 올린다.





4. 느낀점

fruit 사진을 비교해 보았을 때 equalization 과정을 통해 사진의 가시성이 높아졌다고 생각한다. 명암이나 물체의 파악이 더 잘 되었다고 생각한다.