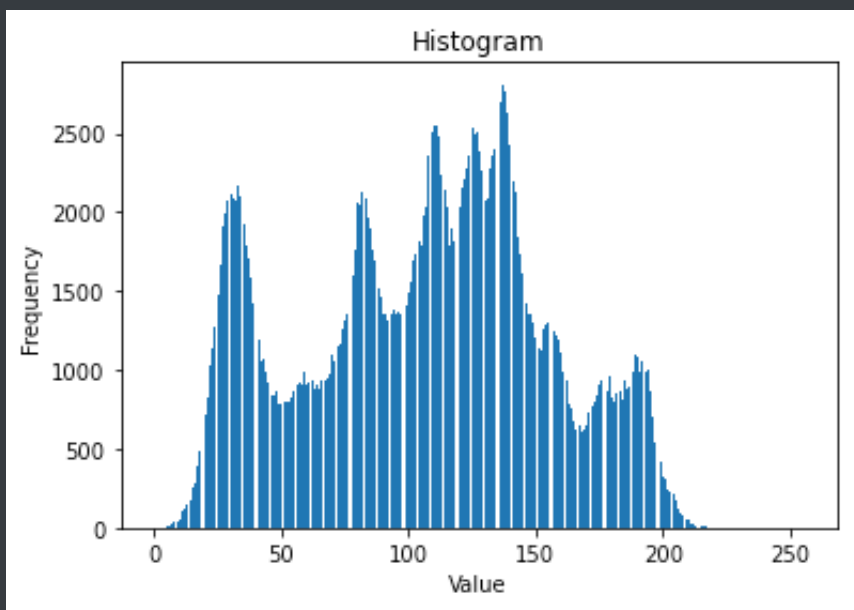


# 멀티미디어정보처리 실습\_0330

소프트웨어학부 2017011803 김예준

## 1. Make\_hist 함수

```
def make_hist(image):  
    assert len(image.shape) == 2, "grayscale 영상을 입력해주세요."  
    histogram = np.zeros((256), dtype=np.int32)  
    h, w = image.shape  
  
    for i in range(h):  
        for j in range(w):  
            # 빈 칸 채우기  
            histogram[image[i,j]] += 1  
    return histogram
```



영상의 전체적인 밝기 특성에 대한 정보를 제공 (밝기 값 제공)

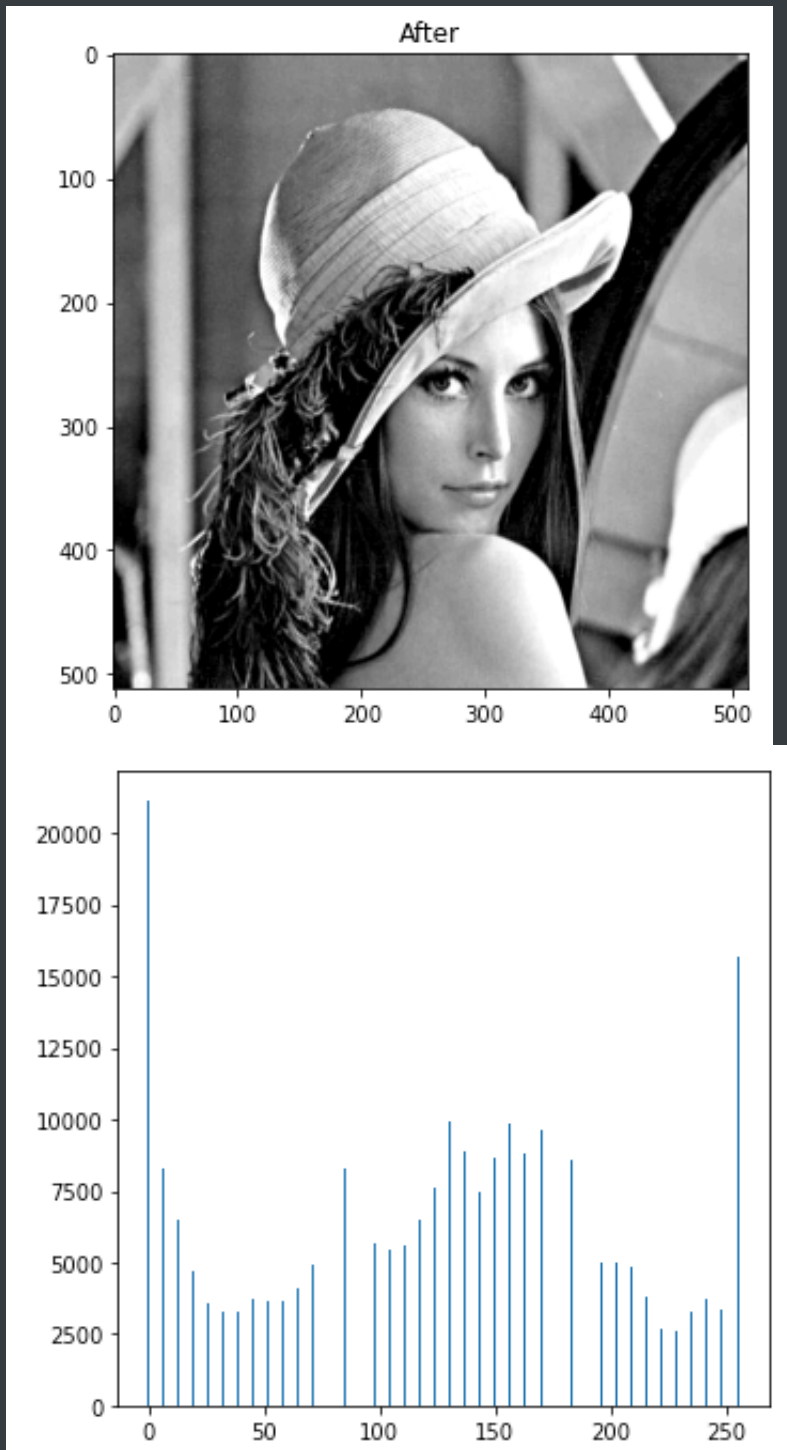
## 2. Histogram Stretching

---

```
slope = 255 / (max_value - min_value)
print("slope: ", slope)
print("(max - min) * slope: ", (max_value - min_value) * slope)

lena_after = np.zeros((h, w), dtype=np.uint8)

for i in range(h):
    for j in range(w):
        #빈 칸 채우기
        if (lena_before[i, j] < min_value):
            lena_after[i, j] = 0
        elif (lena_before[i, j] > max_value):
            lena_after[i, j] = 255
        else:
            lena_after[i, j] = (lena_before[i, j] - min_value) * slope
```



출력 영상의 히스토그램은 모든 범위의 밝기 값을 포함하도록 밝기 분포를 개선

### 3. Histogram. Equalization

```
lena_before = cv2.imread("lena_low.png", 0)
h, w = np.shape(lena_before)
```

```
lena_after = np.zeros((h,w), dtype=np.uint8)

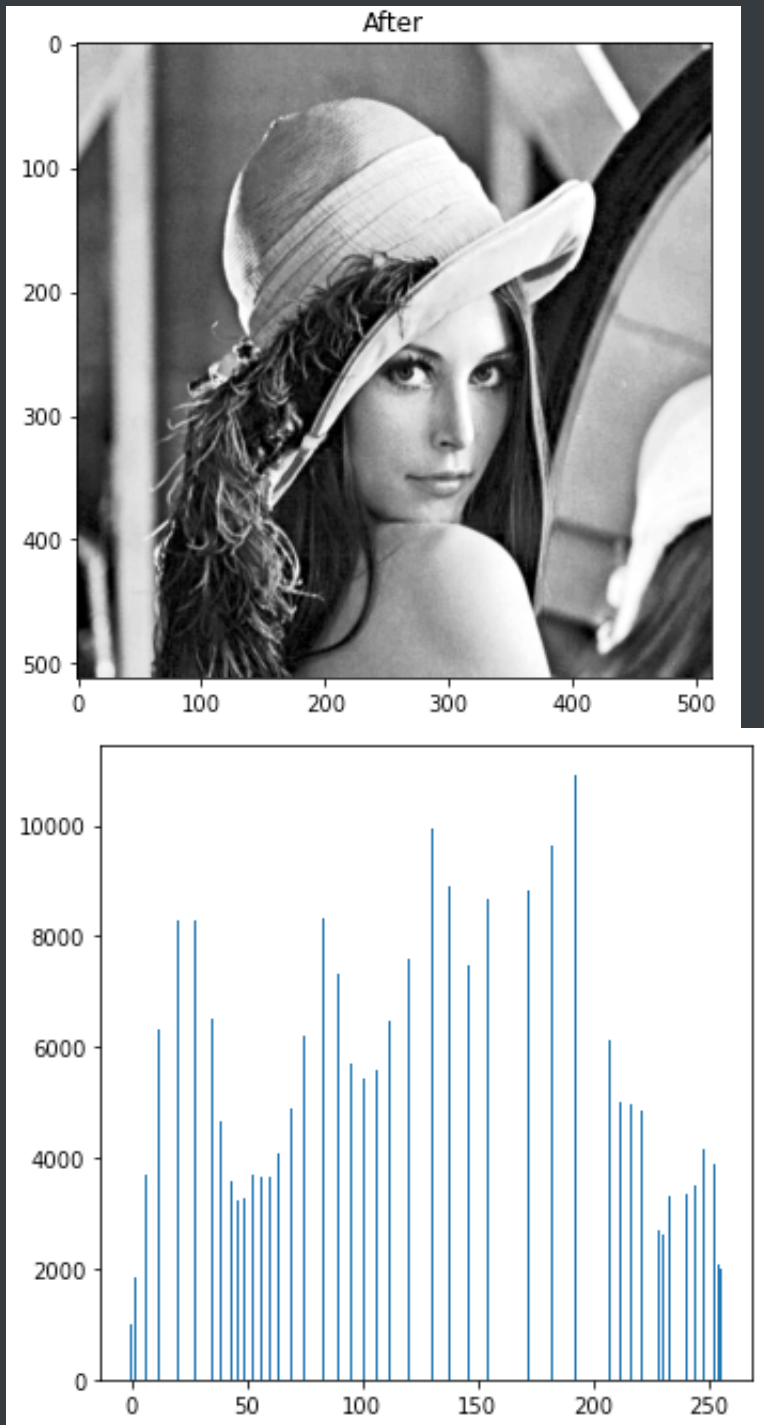
unit = h*w / 256
print("영상크기: ", h, w)
print("unit: ", unit)

lena_hist = make_hist(lena_before)

T=np.zeros((256))
hist_sum = 0
for i in range(256):
    # 빈 칸 채우기
    hist_sum += lena_hist[i]
    T[i] = hist_sum // unit

print("T: ", T)

for i in range(h):
    for j in range(w):
        # 빈 칸 채우기
        lena_after[i,j] = T[lena_before[i,j]]
```



입력 영상의 히스토그램을 균등 분포의 단위 면적(Unit)으로 균일하게 나누어 명암 값 분포를 재분배

## 4. Threshold

```
finger_image = cv2.imread("finger_print.png", 0)
h,w = np.shape(finger_image)

thresh_image = np.zeros((h,w), dtype=np.uint8)
```

```

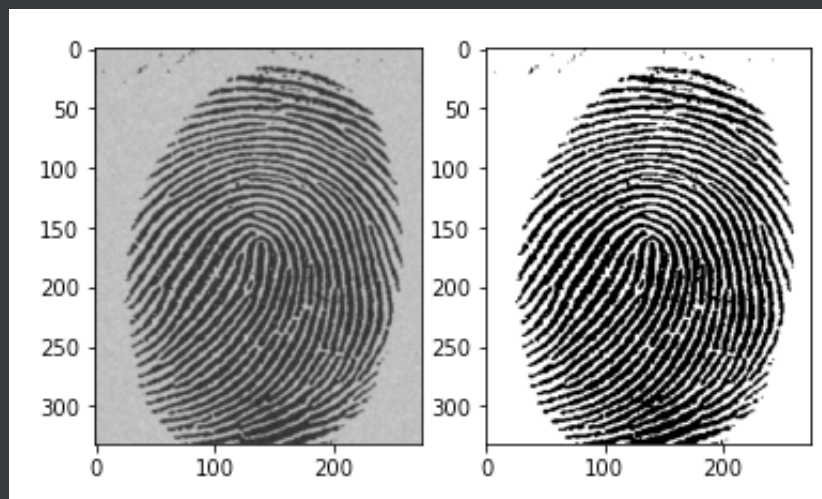
thresh_value = 120

for i in range(h):
    for j in range(w):
        # 빈 칸 채우기
        if (finger_image[i,j] < thresh_value):
            thresh_image[i,j] = 0
        else:
            thresh_image[i,j] = 255

plt.figure()
plt.subplot(1,2,1)
plt.imshow(finger_image, cmap='gray', vmin=0, vmax=255)

plt.subplot(1,2,2)
plt.imshow(thresh_image, cmap='gray', vmin=0, vmax=255)

```



입력 영상의 밝기 값이 특정 값(Threshold)보다 작으면 출력 영상의 밝기 값을 0 크거나 같으면 255로 조정