

# 영상처리 실습 보고서

12주차: Threshold: Otsu's method

학번	201802170
이름	하 상 호

## 1. 과제의 내용

Threshold: Otsu's method 구현

## 2. 과제의 해결 방법

과제의 내용을 해결하기 위해 어떠한 방법을 사용했는지 자세하게 기술한다.

본 과제의 내용을 구현하기 위해선 여러 수학적 과정이 필요하다.

본과정을 수행하기 위해 실습시간에 배운 식들을 이용한다.

$$p_i = n_i / MN \text{ and } \sum_{i=0}^{L-1} p_i = 1$$

히스토그램을 만든 후 전체 합으로 나누어준다.

다만 0으로 나눌 수 있기 때문에 허수를 넣어 error 를 피한다.

```
hist, bins = np.histogram(img.ravel(), 256, [0, 256])  
p = hist / (np.sum(hist) + 1e-7)
```

본 과제의 th 변수는 결국  $k_{opt} = \arg \min_k \sigma_w^2(k)$  이값의 min값이 된다.

arguments of min 즉, 어떤 함수를 최소로 만드는 정의역의 점들, elements 혹은 매개변수를 말한다.

따라서 k\_opt\_varw 변수를 선언하고 이것을 채우기 위해 수행한다.

```

k_opt_varw = []

for k in range(256):

    q1 = p[:k + 1]
    q2 = p[k + 1:]
    q1sum = np.sum(p[:k + 1])
    q2sum = np.sum(p[k + 1:])

    if q1sum == 0 or q2sum == 0:
        k_opt_varw.append(np.inf)
        continue

```

$$q_1(k) = \sum_{i=0}^k p(i)$$

$$q_2(k) = \sum_{i=k+1}^{L-1} p(i)$$

위의 설명 대로 k\_opt\_varw 를 선언하고

. q1 , q2 를 선언해준다.

.q1 또는 q2 가 0일때의 exception 을 만들어 줘야 한다.

이것이 0이 되어 오류가 발생하는 것을 막기 위해 np.inf 를 사용한다.

$$\begin{aligned}
 m_1(k) &= \frac{\sum_{i=0}^k ip(i)}{\sum_{i=0}^k p(i)} = \frac{1}{q_1(k)} \sum_{i=0}^k ip(i) \\
 \sigma_1^2(k) &= \frac{1}{q_1(k)} \sum_{i=0}^k [i - m_1(k)]^2 p(i) \\
 &= \frac{1}{q_1(k)} \sum_{i=0}^k i^2 p(i) - m_1^2(k)
 \end{aligned}$$

$$\begin{aligned}
 m_2(k) &= \frac{\sum_{i=k+1}^{L-1} ip(i)}{\sum_{i=k+1}^{L-1} p(i)} = \frac{1}{q_2(k)} \sum_{i=k+1}^{L-1} ip(i) \\
 \sigma_2^2(k) &= \frac{1}{q_2(k)} \sum_{i=k+1}^{L-1} [i - m_2(k)]^2 p(i) \\
 &= \frac{1}{q_2(k)} \sum_{i=k+1}^{L-1} i^2 p(i) - m_2^2(k)
 \end{aligned}$$

이제 위 과정을 순차적으로 구현해 준다.

```

i1 = [i for i in range(k + 1)]
i2 = [i for i in range(k + 1, 256)]
m1 = np.sum(np.multiply(i1, q1)) / q1sum
m2 = np.sum(np.multiply(i2, q2)) / q2sum

i3 = [i ** 2 for i in range(k + 1)]
i4 = [i ** 2 for i in range(k + 1, 256)]
var1 = np.sum(np.multiply(i3, q1)) / q1sum - np.multiply(m1, m1)
var2 = np.sum(np.multiply(i4, q2)) / q2sum - np.multiply(m2, m2)

```

```

i3 = [i ** 2 for i in range(k + 1)]
i4 = [i ** 2 for i in range(k + 1, 256)]
var1 = np.sum(np.multiply(i3, q1)) / q1sum - np.multiply(m1, m1)
var2 = np.sum(np.multiply(i4, q2)) / q2sum - np.multiply(m2, m2)

```

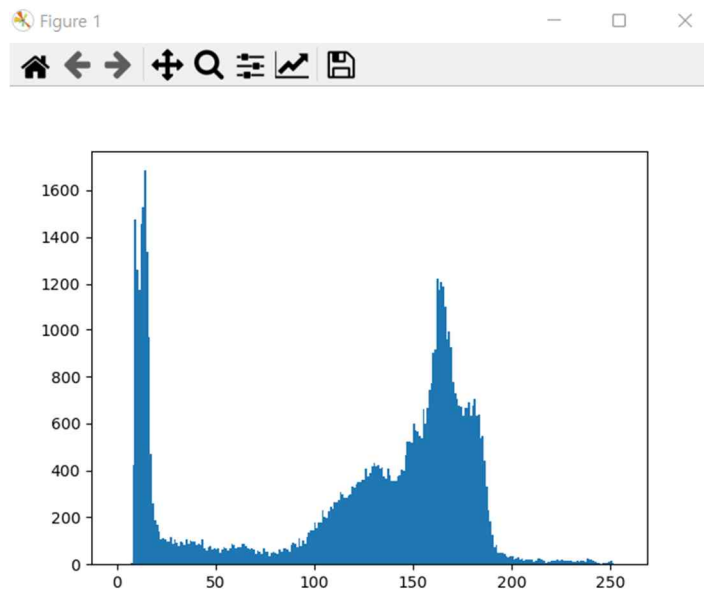
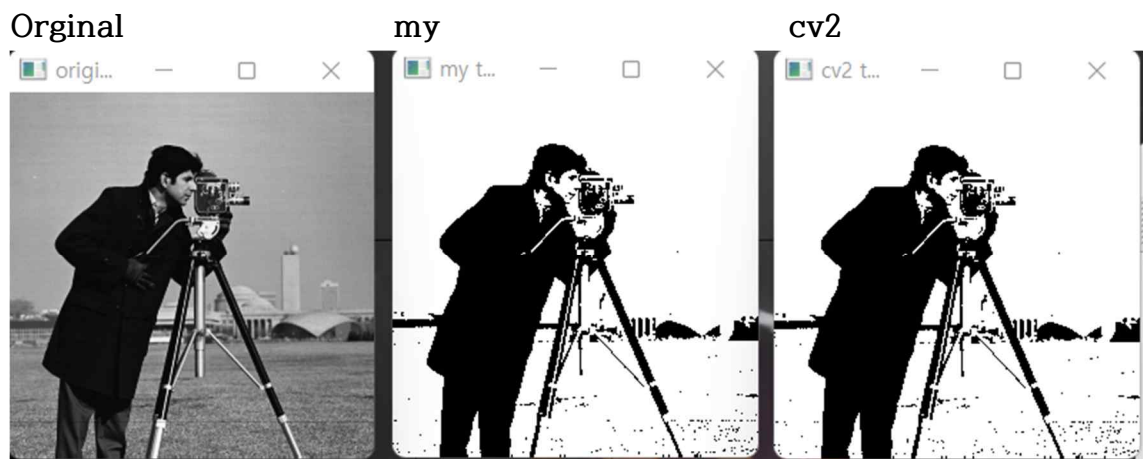
$$\sigma_w^2(k) = q_1(k)\sigma_1^2(k) + q_2(k)\sigma_2^2(k)$$

이제 위부분을 구현 해준다. Varw 를 선언하여 위 부분의 식을 구현 한 후 앞서 선언한 k\_opt\_varw 를 통해서 삽입해준다.

또한 k\_opt\_varw 를 넘파이 배열로 만들어 주고  
Th 변수 에 argmin 을 넣어주면 완료이다.

### 3. 결과물

결과물이 잘 보이도록 화면을 캡처해 보고서에 올린다.



```
C:\Users\sangho\anaconda3\python.exe C:/Users/sangho/OneDrive/Desktop/12주차/otsu_  
Threshold from cv2: 88.0  
Threshold from my: 88
```