

**영상처리**  
**- week9 Compression-1 -**

제출일자	2021.05.16
분 반	01
이 름	강인한
학 번	201701969

```
def Spatial2Frequency_mask(block, n=8):
    dst = np.zeros(block.shape)
    v, u = dst.shape

    y, x = np.mgrid[0:u, 0:v]
    mask = np.zeros((n*n, n*n))
    for v_ in range(v):
        for u_ in range(u):
            tmp = block * np.cos(((2*x+1)*u_*np.pi)/(2*n)) * np.cos(((2*y+1)*v_*np.pi)/(2*n))
            tmp = my_normalize(tmp)
            mask[v_*4:v_*4+4, u_*4:u_*4+4] = tmp
    return mask
```

for문 내부에서 구현한 코드는 다음과 같다.

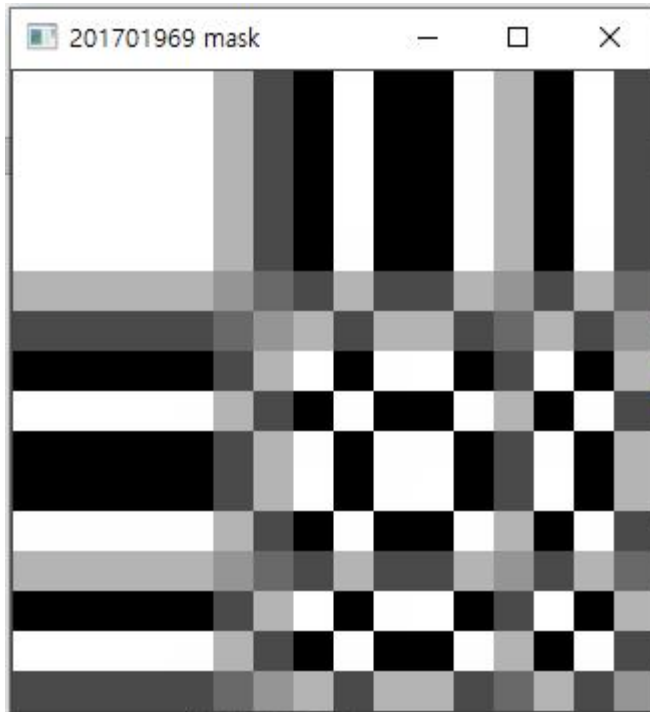
$$f(x, y) \cos\left(\frac{(2x+1)u\pi}{2n}\right) \cos\left(\frac{(2y+1)v\pi}{2n}\right)$$

먼저 위의 식에 대한  $F(0,0)$ 의  $4 \times 4$  마스크를 구현한다. 그 후 상대적 밝기를 정해주기 위해 `my_normalize` 함수를 이용한다. 마지막으로 `mask`의 알맞은 위치로 넣어준다.

코드에서 이중 for문 첫 번째 문장은  $4 \times 4$ 의 밝기가 모두 1인 `block`에  $\cos$  2개를 이용하여 알맞게 바꿔주는 과정이다. 두 번째 `tmp`문은 `my_normalize`를 이용해 상대적 밝기에 따라 밝기를 조절해준다. 3번째 줄은 반환값인 `mask` ( $16 \times 16$ )의 알맞은 위치로 넣어주는 작업이다.

```
def my_normalize(src):
    x = src.copy()
    v, u = src.shape
    if np.sum(x) != v*u*x[0,0]:
        x = x - np.min(x)
    return x / np.max(x) * 255
```

다음으로는 `my_normalize` 함수이다. 상대적 밝기를 통해 밝기를 조절해주는 작업이다. 여기서 주의할점은 모든  $4 \times 4$  공간의 밝기가 모두 같은 경우  $x - \text{np.min}(x) = 0$ 이 되므로 생략해주고 좌표의 밝기가 모두 같지 않은 `src`에 한해서만  $x - \text{np.min}$ 을 해준다.



```
C:\Users\user\PycharmProjects\pythonProject\venv\Scripts\python.exe C:/Us
[[255 255 255 255 255 180 74 0 255 0 0 254 180 0 255 74]
 [255 255 255 255 255 180 74 0 255 0 0 254 180 0 255 74]
 [255 255 255 255 255 180 74 0 255 0 0 254 180 0 255 74]
 [255 255 255 255 255 180 74 0 255 0 0 254 180 0 255 74]
 [255 255 255 255 255 180 74 0 254 0 0 254 180 0 254 74]
 [180 180 180 180 180 149 105 74 180 74 74 180 149 74 180 105]
 [ 74 74 74 74 74 105 149 180 74 180 180 74 105 180 74 149]
 [ 0 0 0 0 0 74 180 255 0 254 255 0 74 255 0 180]
 [255 255 255 255 254 180 74 0 254 0 0 254 180 0 254 74]
 [ 0 0 0 0 0 74 180 254 0 254 254 0 74 254 0 180]
 [ 0 0 0 0 0 74 180 255 0 254 255 0 74 255 0 180]
 [254 254 254 254 254 180 74 0 254 0 0 254 180 0 254 74]
 [180 180 180 180 180 149 105 74 180 74 74 180 149 74 180 105]
 [ 0 0 0 0 0 74 180 255 0 254 255 0 74 255 0 180]
 [255 255 255 255 254 180 74 0 254 0 0 254 180 0 254 74]
 [ 74 74 74 74 74 105 149 180 74 180 180 74 105 180 74 149]]
```

어려운 알고리즘이나 코드를 구성하는 과제는 아니지만, 상대적인 밝기를 조절하는 부분에서 원하는 값이 나오지 않아 시간이 조금 걸렸던 것 같습니다. mask를 다 완성하고 my\_normalize 작업을 해야 한다고 생각해서 그랬던 것 같습니다. 생각을 바꿔 4\*4부분에서 각각 my\_normalize를 하니 값이 잘 나왔습니다.