Image Processing
- morphological -

제출일자	2021.04.29
분 반	01
이 름	강인한
학 번	201701969

1	[구현 코드]

opening과 closing은 각각 dilation과 erosion을 한 번씩 동작하는 것이므로 dilation과 erosion만 짜게 되면 opening과 closing은 쉽게 구현할 수 있다.

```
def dilation(B, S):
    h,w = len(B),len(B[0])
    h_s,w_s = len(S),len(S[0])
    dst= np.zeros((h+2*(h_s//2),w+2*(w_s//2)), dtype=np.uint8)  #더 큰 공간을 만들어 B의 index 오류를 맞아준다

    for row in range(h):
        for col in range(w):
            if B[row][col]==1:
                dst[row:row+h_s,col:col+w_s] = S[:,:]  # 중앙이 1이라면 주변을 모두 1로 바꿔준다.

    dst= dst[h_s//2:h+h_s//2,w_s//2:w+w_s//2]  #B의 크기로 크기 조절을 해준다.
    return dst
```

dilation 함수는 B의 중앙값이 1이라면 주변 S만큼의 공간이 S와 같은 모양이 나오도록 설정해주는 작업이다. index 오류를 해결하기 위해 더 큰 공간을 만든다. (padding과 비슷)

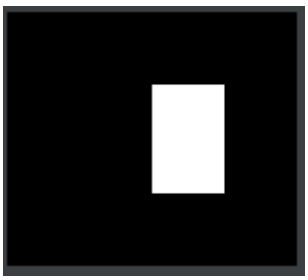
```
def erosion(B, S):
    h,w = len(B),len(B[0])
    h_s,w_s = len(S),len(S[0])
    dst= np.zeros((h+2*(h_s//2),w+2*(w_s//2)), dtype=np.uint8)
    B_plus= np.zeros((h+2*(h_s//2),w+2*(w_s//2)), dtype=np.uint8)

    B_plus[h_s // 2:h + h_s // 2, w_s // 2:w + w_s // 2] = B[:,:]
    for row in range(h):
        for col in range(w):
            if B_plus[row:row+h_s,col:col+w_s].all() == S.all():  #b와 s의 값이 모두 1이라면 중앙 값을 1로 설정한다.
                dst[row+h_s//2,col+w_s//2]=1
    dst = dst[h_s // 2:h + h_s // 2, w_s // 2:w + w_s // 2]  #B의 크기로 다시 조절해준다.
    return dst
```

erosion 함수는 B의 S만큼의 공간이 S와 똑같다면 그 중앙값을 1로 설정해주는 함수이다.

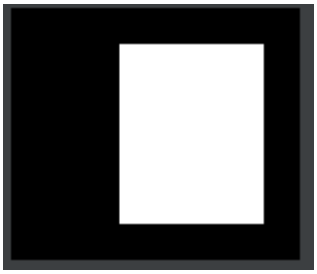
2	[이미지 및 결과]

morphology_dilation

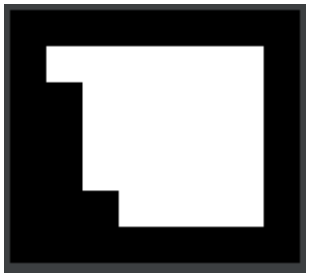
morphology_erosion



morphology_opening



morphology_closing



```
[[255 255 255 255 255 255 255 255]
 [255 255 255 255 255 255 255 255]
 [255 255 255 255 255 255 255 255]
 [  0 255 255 255 255 255 255 255]
 [  0 255 255 255 255 255 255 255]
 [  0 255 255 255 255 255 255 255]
 [  0   0 255 255 255 255 255 255]]
[[  0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0]
 [  0   0   0   0 255 255   0   0]
 [  0   0   0   0 255 255   0   0]
 [  0   0   0   0 255 255   0   0]
 [  0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0]]
[[  0   0   0   0   0   0   0   0]
 [  0   0   0 255 255 255 255   0]
 [  0   0   0 255 255 255 255   0]
 [  0   0   0 255 255 255 255   0]
 [  0   0   0 255 255 255 255   0]
 [  0   0   0 255 255 255 255   0]
 [  0   0   0   0   0   0   0   0]]
[[  0   0   0   0   0   0   0   0]
 [  0 255 255 255 255 255 255   0]
 [  0   0 255 255 255 255 255   0]
 [  0   0 255 255 255 255 255   0]
 [  0   0 255 255 255 255 255   0]
 [  0   0   0 255 255 255 255   0]
 [  0   0   0   0   0   0   0   0]]

Process finished with exit code 0
```

3	[난이도 및 느낀 점]

padding 했을 때와 비슷한 부분이 있다고 느껴져 오래 걸리거나 어렵지 않았던 것 같습니다. 알고리즘 하는 느낌으로 과제를 했던 것 같습니다.