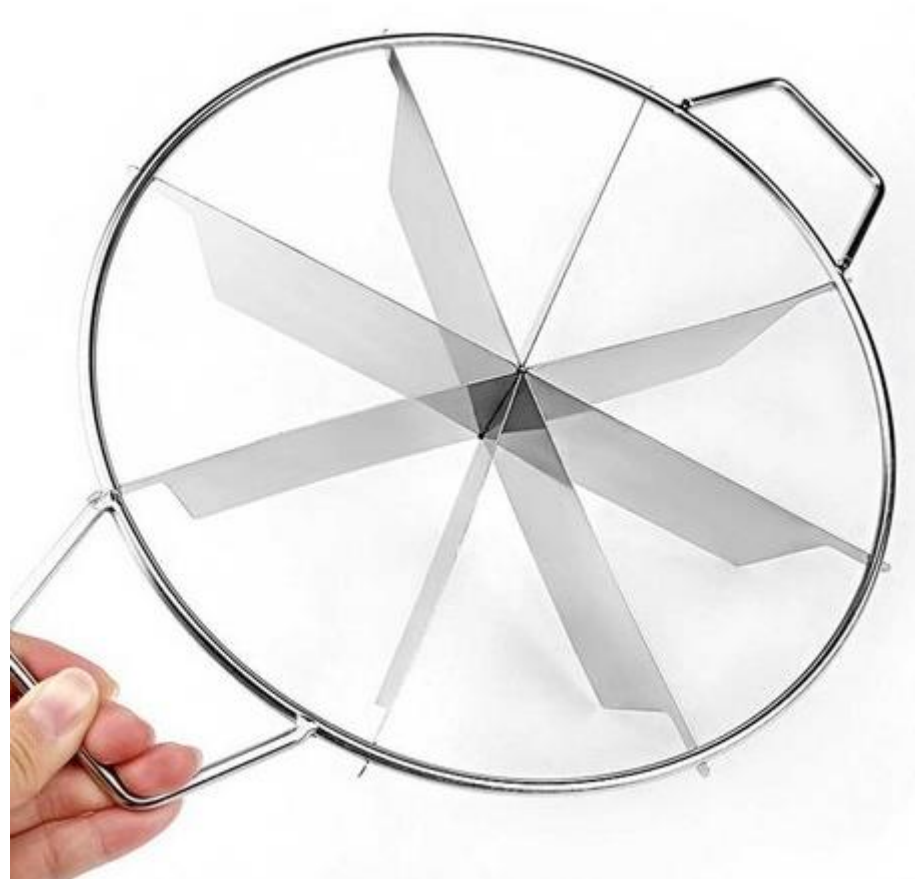


**Image
processing**



AI 명예학회

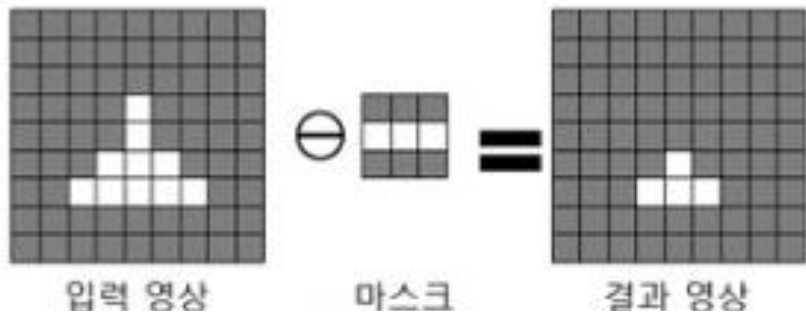
SKHU



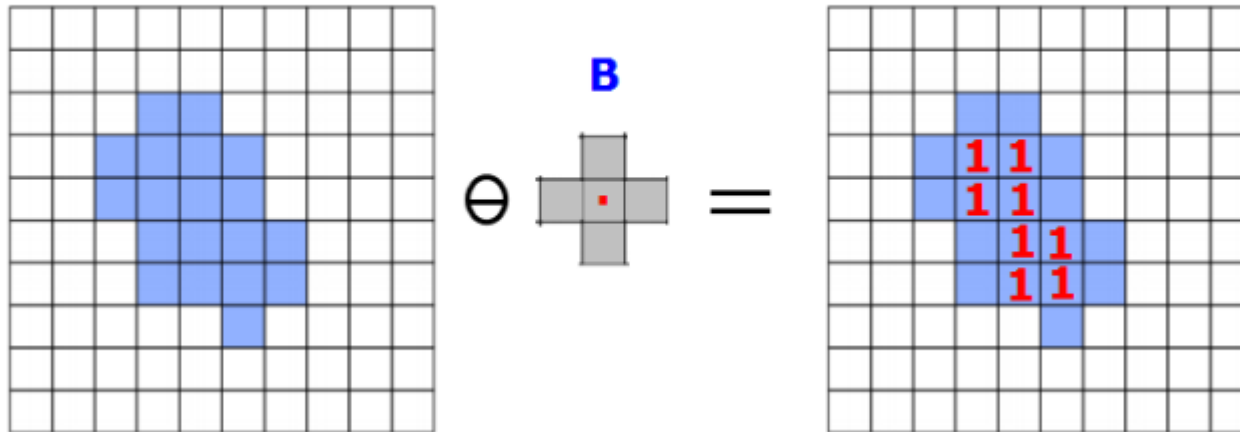
?

모폴로지 연산: 침식(Erosion)

$$A \ominus B$$



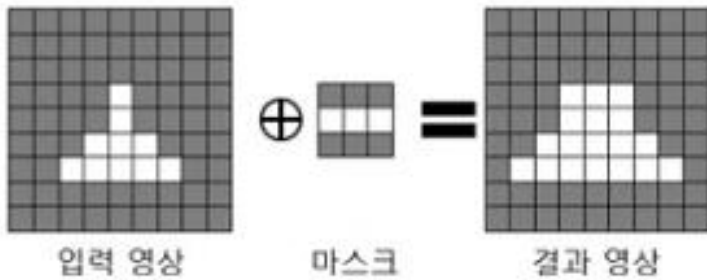
A



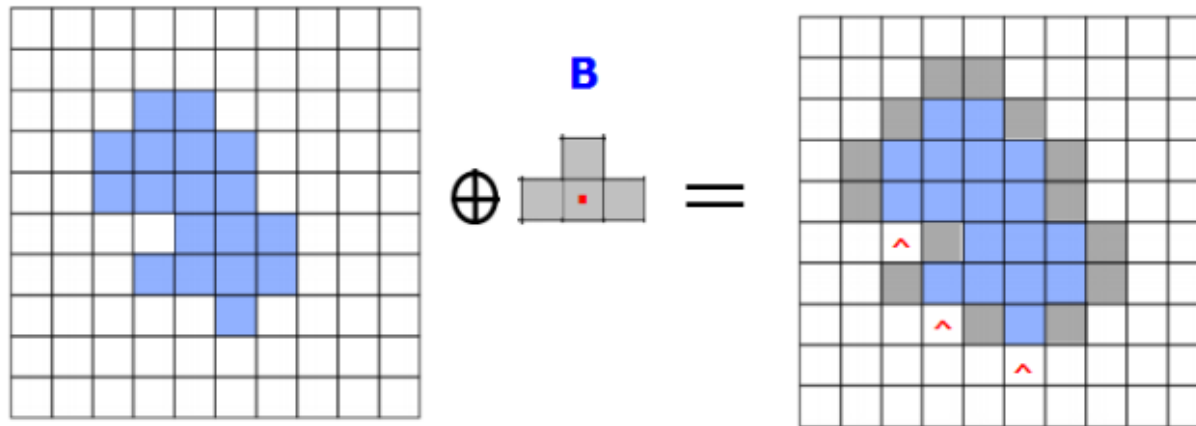
- 노이즈 제거
- 물체 경계 축소
- 물체 내 작은 구멍 제거

모폴로지 연산: 팽창(Dilation)

$$A \oplus B$$



A



- 구멍 채우기
- 물체 연결
- 경계 확장
- But, 노이즈 증가 가능성 존재

모폴로지 연산: 열림(Opening)

$$A \circ B = (A \ominus B) \oplus B$$

1. 작은 객체 제거(침식)
2. 남은 주요 객체 복원(팽창)

노이즈 제거 및 객체 분리

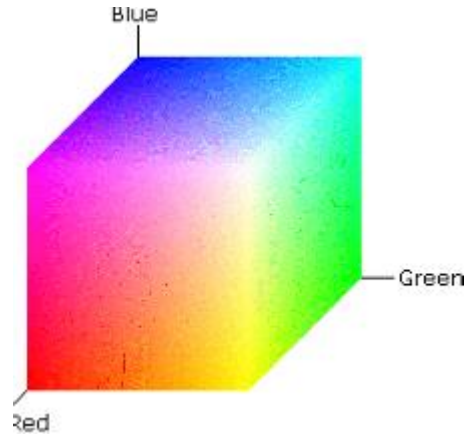
모폴로지 연산: 닫힘(Closing)

$$A \bullet B = (A \oplus B) \ominus B$$

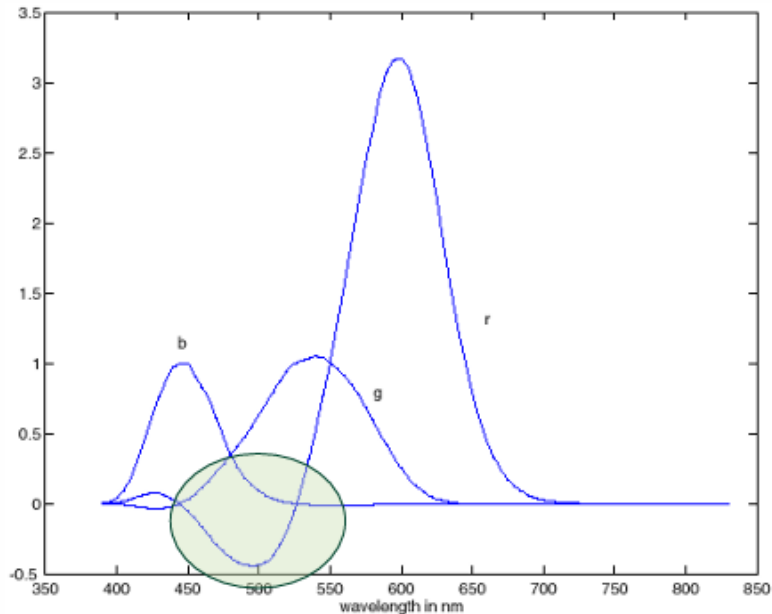
1. 경계 확장(팽창)
2. 경계 보정(침식)

구멍 메우기 및 분리된 객체 조각 연결

Color Space: RGB

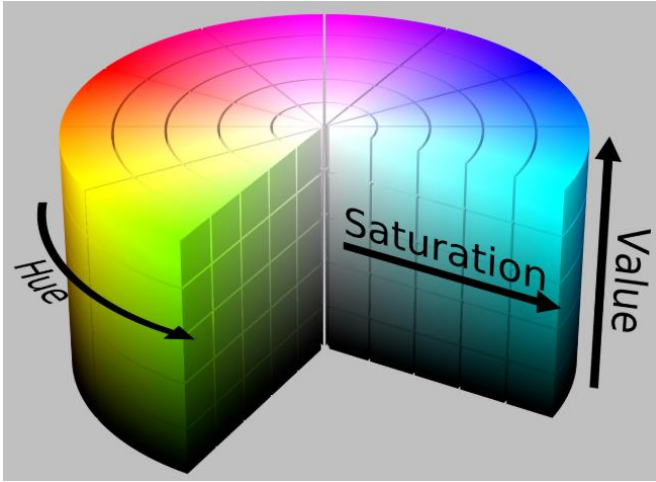


- b: 400~500nm 반응
- g: 500~600nm 반응
- r: 600~700nm 반응



- 빛의 혼합 비율로 색 표현
- 디스플레이 표현 불가능 영역 존재(Negative)

Color Space: HSV



Hue(색상): 색상 종류

Saturation(채도): 색상 선명도

Value(명도): 색 밝기

H,S,V 각각이 Independent



V equalized



SV equalized



HSV equalized

Edge detection



- 밝기 변화 부분 찾는 것
- 최종 물체 경계 추출

Edge detection: Prewitt Operator

-1	-1	-1
1	1	1

수평(G_y)

-1		1
-1		1
-1		1

수직(G_x)

간단하고 빠름.

단순한 구조인 만큼 원본 이미지 밝기 변화 명확해야.

Edge detection: Sobel Operator

-1	-2	-1
1	2	1

수평(G_y)

-1		1
-2		2
-1		1

수직(G_x)

중앙부 값 2값으로 변화.

중앙부 기준 좌우 픽셀 낮은 가중치 부여.

주변 픽셀 덜 중요하게 계산되는 효과. ➔ 잡음제거에 좀 더 나음

Edge detection: Chen-Frei Operator

-1	$-\sqrt{2}$	-1
1	$\sqrt{2}$	1

수평(G_y)

-1		-1
$-\sqrt{2}$		$\sqrt{2}$
1		1

수직(G_x)

중앙부 값 $\sqrt{2}$ 값으로 변화

아까 보다는 좀 더 세밀하게 잡을 수 있다는 장점

중앙부 값 \rightarrow 계산량 증가 단점 존재.

<https://medium.datadriveninvestor.com/understanding-edge-detection-sobel-operator-2aada303b900>

Assignment

1. 수업 내용 정리
2. `basic_segmentation.py` 파일 확인 후,
최대한 눈길 마스킹 잘 되게끔 수정
(HSV 색상 범위, 모폴로지 연산)
3. `edge_detection.py` 파일, 수업 때 배운
Operator로 바꿔서 실습.

제출파일:

- `basic_segmentation.py`
- `edge_detection_Prewitt.py`
- `edge_detection_Sobel.py`