

Python 데이터 분석과 이미지 처리

나동빈

OpenCV Filtering

필터링

- 이미지에 커널을 적용하여 이미지를 흐리게(Blurring = Smoothing) 처리할 수 있습니다.
- 이미지를 흐리게 만들면 노이즈 및 손상을 줄일 수 있습니다.

OpenCV Filtering

컨볼루션 계산

- 특정한 이미지에서 커널(Kernel)을 적용해 컨볼루션 계산하여 필터링을 수행할 수 있습니다.

5	4	3	2	5
4	6	5	4	3
7	5	2	5	9
8	5	6	1	3
5	7	7	8	6

원본 이미지

1	2	1
2	5	2
1	2	1

커널



$6 * 1$	$5 * 2$	$4 * 1$
$5 * 2$	$2 * 5$	$5 * 2$
$5 * 1$	$6 * 2$	$1 * 1$

결과: $68 / 17 = 4$

OpenCV Filtering

Basic Kernel

1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1

$$* \frac{1}{25}$$

Gaussian Kernel

1	4	7	4	1
4	16	26	16	4
7	26	41	26	7
4	16	26	16	4
1	4	7	4	1

$$* \frac{1}{273}$$

OpenCV Filtering

직접 커널을 생성하여 필터 적용하기

```
import cv2
import numpy as np

image = cv2.imread('image.jpg')
cv2.imshow('Image', image)
cv2.waitKey(0)

size = 4
kernel = np.ones((size, size), np.float32) / (size ** 2)
print(kernel)

dst = cv2.filter2D(image, -1, kernel)
cv2.imshow('Image', dst)
cv2.waitKey(0)
```

OpenCV Filtering

Basic Blur

```
import cv2

image = cv2.imread('image.jpg')
cv2.imshow('Image', image)
cv2.waitKey(0)

dst = cv2.blur(image, (4, 4))
cv2.imshow('Image', dst)
cv2.waitKey(0)
```

OpenCV Filtering

Gaussian Blur

```
import cv2

image = cv2.imread('image.jpg')
cv2.imshow('Image', image)
cv2.waitKey(0)

# kernel_size: 홀수
dst = cv2.GaussianBlur(image, (5, 5), 0)
cv2.imshow('Image', dst)
cv2.waitKey(0)
```