

CH13. OpenCV: 이미지 Erosion과 Dilation

13. 이미지 Erosion과 Dilation

- 13.1 테스트 환경

순 번	제 목	설치 버전
1	운영 체제	Windows 10 64bit
2	프로그래밍 언어	Python3.6.5

▪ 13.2 [13-01 example.py] - 이미지 Erosion과 Dilation

- 다음은 13-01 example.py 예제의 code이다.

```
import numpy as np
```

```
import cv2
```

```
def morph() :
```

```
    img = cv2.imread('images/cat.jpg', cv2.IMREAD_GRAYSCALE)
```

```
    kernel = np.ones(3,3), np.uint8)
```

```
    erosion = cv2.erode(img, kernel, iterations=1)
```

```
    dilation = cv2.dilate(img, kernel, iterations=1)
```

▪ 13.2 [13-01 example.py] - 이미지 Erosion과 Dilation

- 다음은 13-01 example.py 예제의 code이다.

```
cv2.imshow('original', img)
cv2.imshow('erosion', erosion)
cv2.imshow('dilation', dilation)
```

```
cv2.waitKey(0)
cv2.destroyAllWindows()
```

```
morph()
```

▪ 13.2 [13-01 example.py] - 이미지 Erosion과 Dilation

- 다음 그림은 13-01 example.py를 실행한 결과 생성되는 image 사진이다.
 뒷 장부터 13-01 example.py code의 주요 함수를 line 순서대로 분석해보도록 한다



[그림] 13-01 example.py의 실행 결과

▪ 13.2 [13-01 example.py] - 이미지 Erosion과 Dilation

>> LINE 7) `kernel = np.ones((3, 3), np.uint8)`

- 3x3 크기의 1로 채워진 매트릭스를 생성한다. 이 매트릭스는 Erosion 및 Dilation을 위한 커널로 사용될 예정이다.

>> LINE 9) `erosion = cv2.erode(img, kernel, iterations=1)`

- 첫 번째 인자 `img` : erosion을 수행할 원본 이미지
두 번째 인자 `kernel`: erosion을 위한 커널
세 번째 인자 `iterations`: Erosion 반복 횟수

>> LINE 10) `dilation = cv2.dilate(img, kernel, iterations=1)`

- `cv2.erode()` 함수와 인자가 동일하지만 이미지를 dilation 한다.

▪ 13.3 [13-02 example.py] - 이미지 Opening과 Closing

- OpenCV에는 Opening과 Closing 이라는 이미지 변형 방법이 있다. 이는 각종 노이즈를 제거하는데 효과적이다.
- Opening 기법은 erosion 수행을 한 후 바로 dialation 수행을 하여 본래 이미지 크기로 돌려 놓는 것이고, Closing 기법은 dialation 수행을 한 후 바로 erosion 수행을 하여 본래 이미지 크기로 돌려 놓는 것이다.

▪ 13.3 [13-02 example.py] - 이미지 Opening과 Closing

- 다음은 13-02 example.py 예제의 code이다.

```
import numpy as np
```

```
import cv2
```

```
def morph() :
```

```
    img1 = cv2.imread('images/opening.jpg', cv2.IMREAD_GRAYSCALE)
```

```
    img2 = cv2.imread('images/closing.jpg', cv2.IMREAD_GRAYSCALE)
```

```
    kernel = np.ones(5, 5), np.uint8)
```


▪ 13.3 [13-02 example.py] - 이미지 Opening과 Closing

- 다음은 13-02 example.py 예제의 code이다.

```
opening = cv2.morphologyEx(img1, cv2.MORPH_OPEN, kernel)
```

```
closing = cv2.morphologyEx(img2, cv2.MORPH_CLOSE, kernel)
```

```
cv2.imshow('opening', opening)
```

```
cv2.imshow('closing', closing)
```

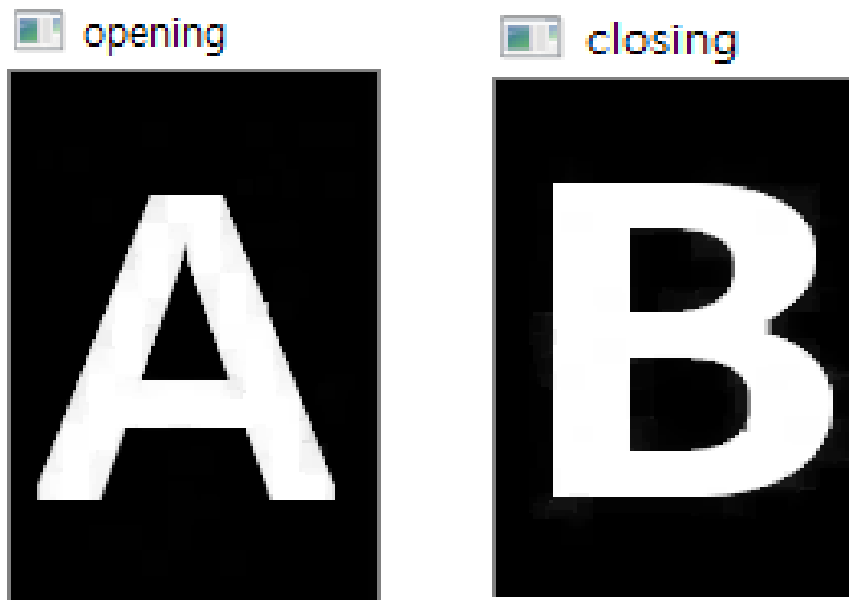
```
cv2.waitKey(0)
```

```
cv2.destroyAllWindows()
```

```
morph()
```

▪ 13.3 [13-02 example.py] - 이미지 Opening과 Closing

- 다음 그림은 13-02 example.py를 실행한 결과 생성되는 image 사진이다.
 뒷 장부터 13-02 example.py code의 주요 함수를 line 순서대로 분석해보도록 한다



[그림] 13-02 example.py의 실행 결과

▪ 13.3 [13-02 example.py] - 이미지 Opening과 Closing

>> LINE 10) `opening = cv2.morphologyEx(img1, cv2.MORPH_OPEN, kernel)`

- `cv2.morphologyEx()` 함수는 erosion과 dilation 알고리즘을 이용해서 보다 향상된 이미지 변형 기능을 제공한다.

- 이 함수의 인자는 다음과 같다.

두 번째 인자 operation: 이미지 변형 오퍼레이션 종류

`cv2.MORPH_OPEN`: Opening을 수행

`cv2.MORPH_CLOSE`: Closing을 수행

`cv2.MORPH_GRADIENT`: Dilation 이미지와 Erosion 이미지의 차이를 나타냄

`cv2.MORPH_TOPHAT`: 원본 이미지와 opening한 이미지의 차이를 나타냄

`cv2.MORPH_BLACKHAT`: closing한 이미지와 원본 이미지의 차이를 나타냄

세 번째 인자 kernel: 적용할 커널 매트릭스

▪ 13.4 [13-03 example.py] - morphologyEx()

- 다음은 13-03 example.py 예제의 code이다.

```
import numpy as np
```

```
import cv2
```

```
def morph() :
```

```
    img1 = cv2.imread('images/alphabet.jpg', cv2.IMREAD_GRAYSCALE)
```

```
    img2 = cv2.imread('images/opening.jpg', cv2.IMREAD_GRAYSCALE)
```

```
    img3 = cv2.imread('images/closing.jpg', cv2.IMREAD_GRAYSCALE)
```

```
    kernel = np.ones(3, 3), np.uint8)
```

▪ 13.4 [13-03 example.py] - morphologyEx()

- 다음은 13-03 example.py 예제의 code이다.

```
grad = cv2.morphologyEx(img1, cv2.MORPH_GRADIENT, kernel)
tophat = cv2.morphologyEx(img2, cv2.MORPH_TOPHAT, kernel)
blackhat = cv2.morphologyEx(img3, cv2.MORPH_BLACKHAT, kernel)
```

```
cv2.imshow('grad', grad)
cv2.imshow('tophat', tophat)
cv2.imshow('blackhat', blackhat)
```

```
cv2.waitKey(0)
cv2.destroyAllWindows()
```

```
morph()
```

▪ 13.4 [13-03 example.py] - morphologyEx()

- 다음 그림은 13-03 example.py를 실행한 결과 생성되는 image 사진이다.
 뒷 장부터 13-03 example.py code의 주요 함수를 line 순서대로 분석해보도록 한다



[그림] 13-03 example.py의 실행 결과

▪ 13.4 [13-03 example.py] - morphologyEx()

- cv2.morphologyEx() 함수에 다음과 같은 내용을 알아보기 위한 코드이다.
 - 1) alp.jpg 이미지의 Dilation 이미지와 Erosion 이미지의 차이를 화면에 디스플레이
 - 2) 백그라운드에 흰색 노이즈가 있는 'A' 글자의 원본 이미지와 opening 처리 후 이미지의 차이를 화면에 디스플레이
 - 3) 포그라운드에 검정색 노이즈가 있는 'B' 글자의 원본 이미지와 closing 처리 후 이미지의 차이를 화면에 디스플레이

▪ 13.5 [13-04 example.py] - 커널 매트릭스 만들기

- 원, 타원 모양의 커널을 만들어 적용해야 할 때, 이러한 매트릭스를 우리가 스스로 생성해서 사용해도 되지만 `cv2.getStructuringElement()` 함수를 이용하면 손쉽게 만들 수 있다.
- 이 함수에 원하는 커널 모양과 커널 크기만 인자로 넘겨주면 이 함수가 커널 매트릭스를 알아서 만들어준다.

▪ 13.5 [13-04 example.py] - 커널 매트릭스 만들기

- 다음은 13-04 example.py 예제의 code이다.

```
import numpy as np
```

```
import cv2
```

```
def makeKernel() :
```

```
    M1 = cv2.getStructuringElement(cv2.MORPH_RECT, (5,5))
```

```
    M2 = cv2.getStructuringElement(cv2.MORPH_ELLIPSE, (5,5))
```

```
    M3 = cv2.getStructuringElement(cv2.MORPH_CROSS, (5,5))
```

▪ 13.5 [13-04 example.py] - 커널 매트릭스 만들기

- 다음은 13-04 example.py 예제의 code이다.

```
print(M1)
```

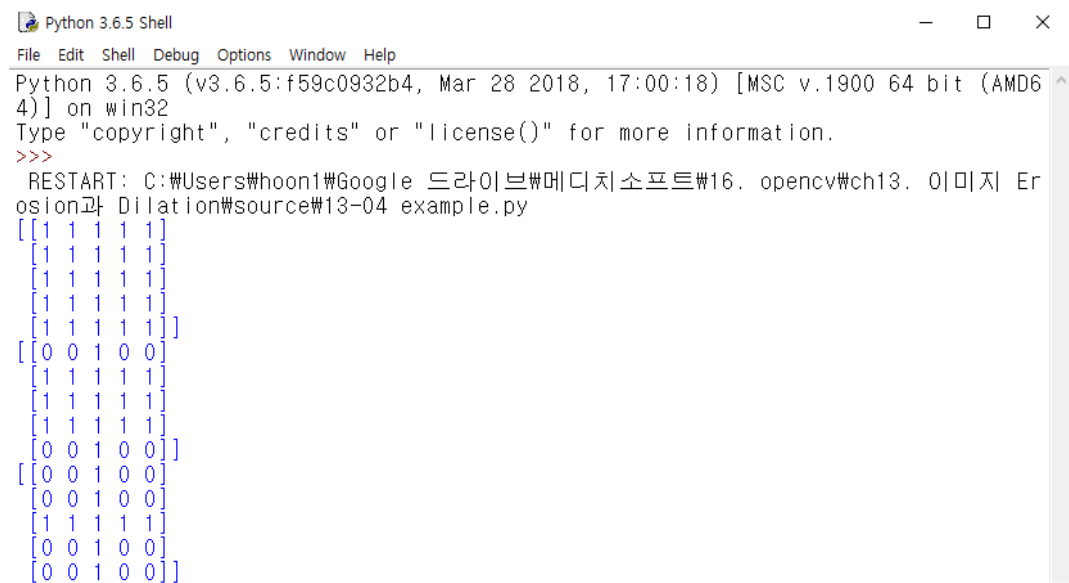
```
print(M2)
```

```
print(M3)
```

```
makeKernel()
```

▪ 13.5 [13-04 example.py] - 커널 매트릭스 만들기

- 다음 그림은 13-04 example.py를 실행한 결과 생성되는 image 사진이다.
 뒷 장부터 13-04 example.py code의 주요 함수를 line 순서대로 분석해보도록 한다



```
Python 3.6.5 Shell
File Edit Shell Debug Options Window Help
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 17:00:18) [MSC v.1900 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
RESTART: C:\Users\whoon1\Google 드라이브\메디치소프트\16. opencv\ch13. 이미지 Erosion과 Dilation\source\13-04 example.py
[[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]
 [0 0 1 0 0]
 [1 1 1 1 1]
 [1 1 1 1 1]
 [1 1 1 1 1]
 [0 0 1 0 0]
 [0 0 1 0 0]
 [0 0 1 0 0]
 [1 1 1 1 1]
 [0 0 1 0 0]
 [0 0 1 0 0]]
```

[그림] 13-04 example.py의 실행 결과

▪ 13.5 [13-04 example.py] - 커널 매트릭스 만들기

```
>> LINE 5) M1 = cv2.getStructuringElement(cv2.MORPH_RECT, (5, 5))
```

- 직사각형 모양으로 5x5 크기의 커널 매트릭스를 생성한다.
- 첫 번째 플래그는 다음 두 가지의 종류가 있다.
 - 1) cv2.MORPH_ELLIPSE: 타원 모양으로 매트릭스를 생성
 - 2) cv2.MORPH_CROSS: 십자 모양으로 매트릭스를 생성

```
import numpy as np
import cv2

def morph() :
    img = cv2.imread('images/alphabet.jpg', cv2.IMREAD_GRAYSCALE)

    kernel = np.ones((3,3), np.uint8)

    erosion = cv2.erode(img, kernel, iterations=1)
    dilation = cv2.dilate(img, kernel, iterations=1)

    cv2.imshow('original', img)
    cv2.imshow('erosion', erosion)
    cv2.imshow('dilation', dilation)

    cv2.waitKey(0)
    cv2.destroyAllWindows()

morph()
```

```
import numpy as np
import cv2

def morph() :
    img1 = cv2.imread('images/opening.jpg', cv2.IMREAD_GRAYSCALE)
    img2 = cv2.imread('images/closing.jpg', cv2.IMREAD_GRAYSCALE)

    kernel = np.ones((5, 5), np.uint8)

    opening = cv2.morphologyEx(img1, cv2.MORPH_OPEN, kernel)
    closing = cv2.morphologyEx(img2, cv2.MORPH_CLOSE, kernel)

    cv2.imshow('opening', opening)
    cv2.imshow('closing', closing)

    cv2.waitKey(0)
    cv2.destroyAllWindows()

morph()
```

```
import numpy as np
import cv2

def morph() :
    img1 = cv2.imread('images/alphabet.jpg', cv2.IMREAD_GRAYSCALE)
    img2 = cv2.imread('images/opening.jpg', cv2.IMREAD_GRAYSCALE)
    img3 = cv2.imread('images/closing.jpg', cv2.IMREAD_GRAYSCALE)

    kernel = np.ones((3, 3), np.uint8)

    grad = cv2.morphologyEx(img1, cv2.MORPH_GRADIENT, kernel)
    tophat = cv2.morphologyEx(img2, cv2.MORPH_TOPHAT, kernel)
    blackhat = cv2.morphologyEx(img3, cv2.MORPH_BLACKHAT, kernel)

    cv2.imshow('grad', grad)
    cv2.imshow('tophat', tophat)
    cv2.imshow('blackhat', blackhat)

    cv2.waitKey(0)
    cv2.destroyAllWindows()

morph()
```

```
import numpy as np
import cv2

def makeKernel() :
    M1 = cv2.getStructuringElement(cv2.MORPH_RECT, (5,5))
    M2 = cv2.getStructuringElement(cv2.MORPH_ELLIPSE, (5,5))
    M3 = cv2.getStructuringElement(cv2.MORPH_CROSS, (5,5))

    print(M1)
    print(M2)
    print(M3)

makeKernel()
```