

CH07. OpenCV: 이미지 픽셀 조작 및 ROI 조작

7. 이미지 픽셀 조작 및 ROI 조작

- 7.1 테스트 환경

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

7. 이미지 픽셀 조작 및 ROI 조작

- 7.2 [07-01 example.py] – 이미지 픽셀 값 얻고 수정하기(1)
 - 다음은 07-01 example.py 예제의 code이다.

[그림] 07-01 example.py의 code

7. 이미지 픽셀 조작 및 ROI 조작

- 7.2 [07-01 example.py] – 이미지 픽셀 값 얻고 수정하기(1)
 - 다음은 07-01 example.py 예제의 code와 실행 결과로 나오는 화면이다. 뒷장 부터 07-01 example.py code의 주요 함수를 line 순서대로 분석해보도록 한다.

[그림] 07-01 example.py의 실행 결과

7. 이미지 픽셀 조작 및 ROI 조작

▪ 7.2 [07-01 example.py] – 이미지 픽셀 값 얻고 수정하기(1)

```
>> LINE 3) img = cv2.imread('images/hallstatt.jpg')
```

```
>> LINE 4) px = img[340, 200]
```

```
>> LINE 5) print(px)
```

- 위 코드는 images 폴더에 있는 hallstatt.jpg 파일을 읽어, (340, 200) 위치의 픽셀값을 출력하는 코드이다. 이미지에 따라서 다르겠지만 출력값은 [69, 76, 61]와 같이 표시된다.
이는 hallstatt.jpg의 (340, 200)위치에 있는 픽셀은 Blue = 69, Green = 76, Red = 61 인 색상을 의미한다.

```
>> 참고 사항) img[340, 200] = [0, 0, 0]
```

- 만약 이 픽셀을 검정으로 바꾸고 싶다면 아래의 코드를 적용한다.

7. 이미지 픽셀 조작 및 ROI 조작

- 7.3 [07-02 example.py] – 이미지 픽셀 값 얻고 수정하기(2)
 - 다음은 07-02 example.py 예제의 code이다.

[그림] 07-02 example.py의 code

[그림] 07-02 example.py의 실행 결과

7. 이미지 픽셀 조작 및 ROI 조작

▪ 7.3 [07-02 example.py] – 이미지 픽셀 값 얻고 수정하기(2)

- 다음은 07-02 example.py 예제의 code와 실행 결과로 나오는 화면이다. 뒷장 부터 07-02 example.py code의 주요 함수를 line 순서대로 분석해보도록 한다.
- 07-01 example.py 예제는 개개의 픽셀 작업을 수행하는데 있어 성능상 문제가 있을 수 있다. Numpy는 이러한 계산에 최적화된 알고리즘을 적용하였으므로 개별적인 픽셀에 접근하여 그 값을 변경하려면 07-02 example.py의 코드를 사용하는게 낫다.

7. 이미지 픽셀 조작 및 ROI 조작

▪ 7.3 [07-02 example.py] – 이미지 픽셀 값 얻고 수정하기(2)

```
>> LINE 6) B = img.item(340, 200, 0)
```

```
>> LINE 7) G = img.item(340, 200, 1)
```

```
>> LINE 8) R = img.item(340, 200, 2)
```

- Numpy array의 item() 함수는 개별적인 픽셀에 접근할 수 있지만, B, G, R 개별적으로 접근해야 한다. 만약 (340, 200) 위치의 픽셀값을 변경하려면 img.itemset() 함수를 이용하면 되고, 이 역시 B, G, R 개별적인 값을 변경해주어야 한다.

```
>> 참고 사항) img.itemset((340, 200, 0), 100)
```

- (340, 200) 위치의 픽셀의 Blue 값을 100으로 변경하는 코드이다.

7. 이미지 픽셀 조작 및 ROI 조작

- 7.4 [07-03 example.py] – 이미지 속성 얻기
 - 다음은 07-03 example.py 예제의 code이다.

[그림] 07-03 example.py의 code

[그림] 07-03 example.py의 실행 결과

7. 이미지 픽셀 조작 및 ROI 조작

- 7.4 [07-03 example.py] – 이미지 속성 얻기

- 이미지 속성을 얻기 위해서는 Numpy array의 아래와 같은 값들에 접근하면 된다.
 - (1) `img.shape` : 이미지 해상도 및 컬러 채널 (이미지 height, 이미지 width, 컬러 채널수)
 - (2) `img.size` : 이미지 사이즈(바이트)
 - (3) `img.dtype` : 이미지 데이터 타입

7. 이미지 픽셀 조작 및 ROI 조작

- 7.5 [07-04 example.py] – ROI(Region Of Image)는 Numpy 인덱싱을 통해 얻을 수 있다.
 - 다음은 07-04 example.py 예제의 code이다.

[그림] 07-04 example.py의 code (1/2)

[그림] 07-04 example.py의 code (2/2)

```
import numpy as np
import cv2

img = cv2.imread('images/hallstatt.jpg')
px = img[340, 200]
print(px)
```

```
import numpy as np
import cv2

img = cv2.imread('images/hallstatt.jpg')

B = img.item(340, 200, 0)
G = img.item(340, 200, 1)
R = img.item(340, 200, 2)

BGR = [B, G, R]
print(BGR)
```

```
import numpy as np
import cv2

img = cv2.imread('images/hallstatt.jpg')

print(img.shape)
print(img.size)
print(img.dtype)
```