

HW 01 – REPORT

소속 : 정보컴퓨터공학부

학번 : 202255595

이름 : 임영훈

1. 서론

이번 실습의 목표는 파이썬을 이용하여 이미지를 핸들링 해보는 것이다.

이미지 핸들링을 위해 pillow(PIL)라는 라이브러리를 사용한다.

2. 본론

1. 이미지 열기

먼저 이미지를 불러와야 한다. 이미지를 불러오기 위해 **Image.open('이미지. 경로')** 함수를 실행한다

```
im = Image.open('chipmunk.png')
```

이 Image class는 size, mode, format 등 여러 attribute를 가진다.

```
print (im.size, im.mode, im.format)
```

 실행시

```
(750, 599) RGB JPEG
```

 출력

2. 이미지 보기

PIL은 이미지를 보여주는 내장된 도구가 없다. 하지만 Open 한 이미지는 **show()** method로 볼 수 있는데, 이는 디버깅을 위해 이미지를 디스크의 임시 파일에 저장하고 플랫폼에 따른 외부 display 유틸리티를 호출하기 때문이다.(유닉스의 경우 xv, 윈도우의 경우 Paint 프로그램)

```
im.show()
```

만약 제대로 동작하지 않을시 matplotlib의 pyplot 모듈을 사용하여 볼 수 있다

3. 이미지 조작

불러온 이미지를 변형 시키기 위해 **convert()** method를 사용 할 수 있다

이미지를 흑백으로 변경을 원한다면 파라미터로 'L'을 넣으면 된다.('RGB', 'P' 등도 가능)

```
im = im.convert('L')
```

만약 이미지의 특정 부분만 가져오고 싶다면 **crop()** method를 사용할 수 있다. 파라미터로는 가져오길 원하는 범위를 넣어주면 된다.

```
im2 = im.crop((280,150,430,300))
```



4. 이미지 저장

```
im2.save('chipmunk_head.png', 'PNG')
```

새롭게 만든 이미지를 save() method를 통해 저장 가능하다

5. numpy 활용

PIL과 numpy는 내부적으로 다른 표현방식을 사용한다. np.asarray(im)을 사용하면 numpy array 형태의 이미지를 얻을 수 있다

```
im2_array = np.asarray(im2)
```

```
[[ 55  55  55 ...  57  57  56]
 [ 54  54  54 ...  57  56  55]
 [ 55  55  55 ...  60  56  58]
 ...
 [101  79 103 ... 115 115 129]
 [ 98  83 106 ... 116 117 114]
 [ 97  89 109 ... 113 120 102]]
```

이렇게 intensity(강도)로 이루어진 배열을 얻을 수 있는데 강도는 0~255의 범위를 가지고 강도가 클수록 밝다. 전체 pixel에 intensity를 더하면 밝은 이미지를 얻을 수 있다

```
for x in range(0,150):
    for y in range(0,150):
        im3_array[y,x] = min(im3_array[y,x] + 50, 255)
```

Numpy 배열에서 다시 Image로 바꾸기 위해서는 Image.fromarray(배열) 을 사용하면 된다.

```
im3 = Image.fromarray(im3_array)
```

Intensity를 줄이면 반대로 어두운 이미지를 얻을 수 있는데 이때 numpy 배열에 0.5를 곱하면 원

래보다 작은 intensity를 가진 배열을 얻을 수 있지만 0~255의 정수형에서 벗어난 실수 형 값을 가질 수 있기 때문에 이를 다시 정수 값으로 바꾸어 주어야 하고 이때 astype method를 사용할 수 있다. `im4_array = im4_array.astype('uint8')` (여기서 uint8은 부호가 없는 8bit 정수를 의미함)



6. 이미지 만들기

Intensity가 담긴 배열을 만들어 이미지를 만들 수도 있다.

```
grad = np.arange(0,256)
```

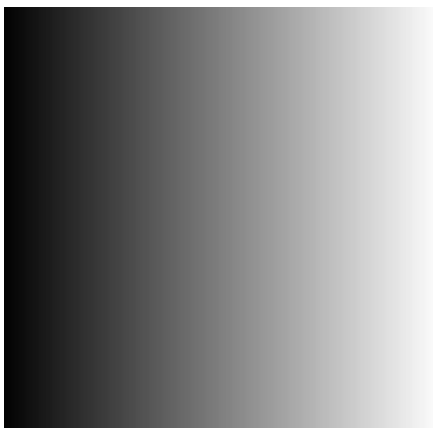
[0, 1, 2, ... 254, 255]의 1차원 배열을 만든다

```
grad = np.tile(grad,[256,1])
```

위에서 만든 배열을 tile(배열, 형태) 를 통해 쌓아준다.

```
im5 = Image.fromarray(grad.astype('uint8'))  
im5.save('gradient.png','PNG')
```

이를 이미지 형태로 바꾸면 새로운 이미지를 얻을 수 있다.



3. 결론

파이썬의 pillow 라이브러리를 통하여 이미지를 불러와 핸들링 할 수 있고 numpy를 통해 이미지를 숫자(intensity)로 바꾸어 좀 더 직접적으로 조정이 가능하다는 것을 알았고 intensity가 담긴 배열을 만들어 자신만의 이미지를 만들 수 있음을 알 수 있었다.