

# HW 01 – REPORT

소속 : 정보컴퓨터공학부

학번 : 201924548

이름 : 이풍헌

# 1. 서론

실습 목표 및 이론적 배경 기술 (1~2페이지)

이번 실습의 목표는 파이썬의 PIL과 numpy 라이브러리를 사용하여 디지털 이미지를 처리해보는 것이다. 이미지 파일을 열고, 그레이 스케일로 변환 후 sub-region을 crop해 밝기 조작을 해보고 새로운 이미지를 생성하고자 한다.

그레이 스케일 이미지는 픽셀당 1byte를 할당해 0~255의 값을 가지며 0은 black, 255는 화이트의 색상을 갖게된다. 이미지는 픽셀의 2차원 배열로 나타내어지며 이 배열의 값을 변경함으로써 이미지를 변경할 수 있다.

실습에 사용할 라이브러리 PIL은 python imaging library의 약자로 다양한 이미지 형식을 지원하고 처리할 수 있는 라이브러리이다. Local의 이미지를 불러와 볼 수 있고, 그레이 스케일로의 변환, 크기 자르기, 배열을 이미지로 전환, local에 사진 저장 등의 기능을 수행할 수 있다.

Numpy는 파이썬 리스트보다 더욱 효율적으로 배열을 처리할 수 있도록 도와주는 라이브러리이다. 자주 사용하는 배열의 최대값, 최소값, 평균 등 다양한 기능을 지원한다.

# 2. 본론

실습 내용 및 결과 기술 (2페이지 이상)

PIL 라이브러리를 통해 사진을 읽어 im변수에 저장한다. im의 size mode format을 통해 크기, 색 모드, 형식 등을 확인할 수 있다. 그후 convert를 통해 luminance 그레이 스케일 이미지로 변환한다.

```
im = Image.open('chipmunk.png')
print (im.size, im.mode, im.format)
im = im.convert('L')
```

Chipmunk\_head.png

다람쥐의 머리부분만 필요로 하는 사진을 얻기 위해 Image의 crop기능을 사용해 원하는 부분의 사진만 얻을 수 있다. 이때 매개변수로는 좌측상단 x, y좌표 우측하단 x, y좌표를 tuple로 넘겨야 한다. 이후 im2에 할당된 잘라낸 사진을 save를 통해 새로운 이미지로 저장한다.

```
im = im.convert('L')
im2 = im.crop((280,150,430,300))
im2.save('chipmunk_head.png', 'PNG')
```



Chipmunk\_head.png

Chipmunk\_head\_bright.png

PIL과 numpy는 다른 표현법을 사용하기 때문에 위에서 만든 사진을 사용하기 위해 numpy 라이브러리를 사용해 array로 만들어준다. im3\_array에 im2\_array를 copy해 모든 원소 값에 50을 추가해 주었다. 그레이 스케일의 경우 숫자가 클수록 밝은 색을 의미하기에 사진을 더 밝게 변경하였다.

```
im2_array = np.asarray(im2)
for x in range(0,150):
    for y in range(0,150):
        im3_array[y,x] = min(im3_array[y,x] + 50, 255)
im3 = Image.fromarray(im3_array)
im3.save('chipmunk_head_bright.png', 'PNG')
```



Chipmunk\_head\_bright.png

Chipmunk\_head\_dark.png

마찬가지로 사진을 어둡게 하기위해 원소의 값을 절반으로 줄였다. 이 과정에서 배열의 원소 값이 실수로 변하게 되기에 사진으로 저장하기위해 type을 다시 정수형으로 변경해주어야 한다.

```
im4_array = im4_array * 0.5
im4_array = im4_array.astype('uint8')
im4 = Image.fromarray(im4_array)
im4.save('chipmunk_head_dark.png', 'PNG')
```

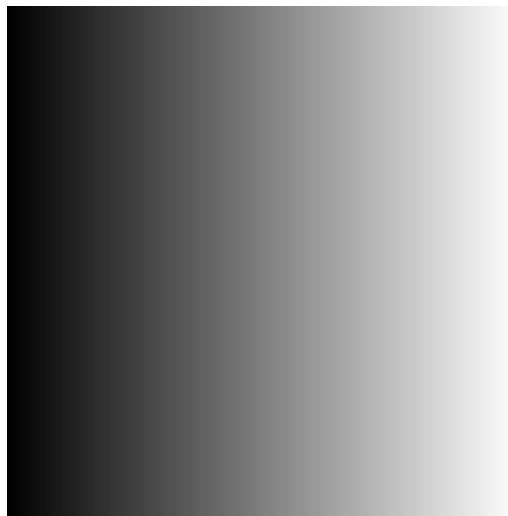


Chipmunk\_head\_dark.png

### Gradient.png

새로운 이미지를 만들기 위해 `np.arange`를 사용해 새로운 1차원 배열을 만들어준다. 이 배열의 값은 0부터 255까지 순차적으로 만들어져 있다. 이 1차원 배열을 다시 256번 사용하여 정사각형 모양의 2차원 배열을 생성한다. 그 후 정수형으로 바꾼 배열로부터 이미지를 생성해 저장하면 왼쪽에서 오른쪽으로 밝아지는 `gradient`이미지가 생성된다.

```
grad = np.arange(0,256)
grad = np.tile(grad,[256,1])
im5 = Image.fromarray(grad.astype('uint8'))
im5.save('gradient.png','PNG')
```



Gradient.png

### 3. 결론

토의 및 결론 (1페이지)

기초적인 이미지 처리를 실습하였다. local 이미지를 불러와 처리하는 PIL 라이브러리와 배열을 처리하는 Numpy 라이브러리를 사용하였다. 다람쥐 사진에 원하는 부분을 뽑아내 간단한 filtering을 통해 이미지를 밝게 또는 어둡게 변환하였다. 실습에서 kernel size를 1로 하여 각 픽셀의 정보가 하나의 픽셀에만 적용되었다. 이후 numpy를 이용한 배열을 통해 새로운 이미지를 생성하는 방법을 배웠다. 이번 실습을 통해 라이브러리를 활용하여 python을 통해 기초적인 이미지를 다루는 방법을 알게 되었다.