

# HW 01 – REPORT

소속 : 정보컴퓨터공학부

학번 : 2021925111

이름 : 김건호

# 1. 서론

2024 컴퓨터비전개론 첫 번째 과제에 대한 보고서입니다. Conda 가상환경에서, Pillow와 numpy library를 통해 이미지의 intensity를 조절하는 방법을 배웁니다. Pillow library는 오픈소스 기반으로 이미지 처리와 그래픽 기능을 제공하는 라이브러리고, numpy library는 numeric python library의 줄임말로, 수치해석에 사용되는 라이브러리입니다. 두 라이브러리에 대한 기초적인 사용법 및, 이를 통해 간단한 이미지 intensity 조절을 하는 것이 목표입니다.

## 2. 본론

PIL 라이브러리를 사용해서 PIL형태로 이미지를 불러옵니다.

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

PIL 라이브러리는 따로 image를 display하는 tool이 없습니다. 따라서, 디버깅을 위해 show method를 사용해서 이미지를 확인할 수 있습니다. 또한 convert method에 매개변수로 'L'을 사용해서 흑백 이미지로 변환합니다. 매개변수로 'RGB'를 주게 된다면, 흑백 이미지가 아닌, RGB 이미지로 변환됩니다. Crop method를 사용해서 다람쥐의 머리 부분에 해당하는 이미지를 자른 후, im2 save method를 통해서 PNG 파일로 저장합니다.

```
# display the image
im.show()
im = im.convert('L')
im2 = im.crop((280,150,430,300))
# save the selected region
im2.save('chipmunk_head.png', 'PNG')
'chipmunk_head.png'
```



im2에 저장되어 있는 흑백 이미지를 numpy 라이브러리를 사용해, im2\_array 변수에, numpy.ndarray type으로 저장합니다. 이후 흑백 이미지의 평균값을 average 변수에 저장합니다.

```
im2_array = np.asarray(im2)
average = np.mean(im2_array)
print("average : ", average)
```

125.47이라는 평균값을 얻을 수 있었습니다. 이는 검은색과 흰색의 중간보다 아주 조금 밝은 정도의 값입니다.

```
average : 125.47693333333333
```

im2\_array 150x150 흑백 이미지를 deep copy 한 후, 밝기를 50 올려줍니다. Intensity는 unit8(0~255 사이 값)으로 저장되는데, 밝기를 50 올린 결과가 255를 초과할 수 있으므로, min 함수를 사용해서 상한을 조절합니다. 이를 fromarray method를 사용해서 numpy 배열을 PIL형태로 저장한 후, save method를 통해 'chipmunk\_head\_bright.png'파일로 저장합니다.

```
im3_array = im2_array.copy()
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'
```



im2\_array 150x150 흑백 이미지를 deep copy 한 후, 각 pixel의 밝기를 반으로 줄입니다. Im2\_array는 numpy.ndarray type이기 때문에, 0.5를 곱하는 연산을 할 수 있습니다. 0.5를 곱했기에, 실수형을 포함하고 있습니다. 이를 astype method를 사용해서, unit8형태의 정수로 다시 저장한 후, fromarray method를 사용해서 PIL형태로 이미지를 저장합니다. 밝기에 0.5를 곱했기에, 기존의 chipmunk\_head.png보다 어두워짐으로 예상됩니다.

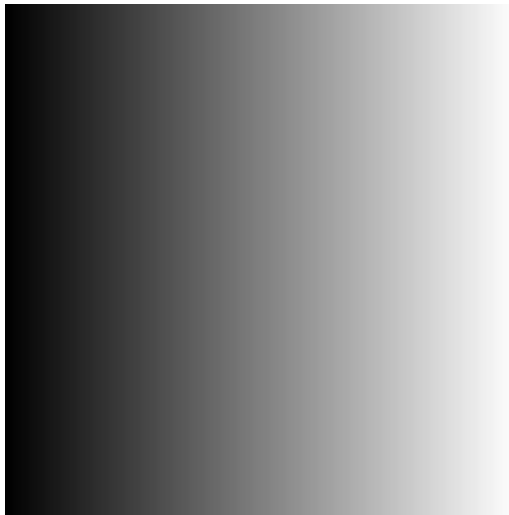
```
im4_array = im2_array.copy()
im4_array = im4_array * 0.5
im4_array = im4_array.astype('uint8')
im4 = Image.fromarray(im4_array)
im4.save('chipmunk_head_dark.png', 'PNG')
'chipmunk_head_dark.png'
```



Arrange method를 사용해서 0~255값을 ndarray에 저장합니다. Tile method를 사용해서 256x256 size의 2차원 ndarray를 형성합니다. 맨 앞부분은 검은색이고, 점점 갈수록 밝기가 밝아지는 gradient image file을 얻을 수 있을 것이라 예상됩니다. Fromarray method와 astype method를 사

용해서 PIL형태로 이미지를 저장한 후, save method를 사용해 이미지 파일을 png 파일로 저장합니다.

```
grad = np.arange(0,256) #numpy.darray 형태로 0~255 값 저장
grad = np.tile(grad,[256,1])
# convert to uint8 and then to a PIL image and save
im5 = Image.fromarray(grad.astype('uint8'))
im5.save('gradient.png','PNG')
'gradient.png'
```



### 3. 결론

Python 내장 자료구조 list를 사용하지 않고, numpy ndarray를 사용해서 intensity 변경을 쉽게 할 수 있었습니다. 또한, convert method가 꼭 이미지의 흑백 intensity만을 조절하는 것이 아닌, 'RGB' 매개변수를 사용해서 image의 RGB값을 조절할 수 있다고 생각할 수 있었습니다. 아래는 RGB 매개변수를 사용한 결과물입니다. 이번 과제에서 이미지 처리 라이브러리 pillow를 사용해서, image의 crop와 intensity 조절하는 방법을 학습할 수 있었습니다. 또한 이를 응용하여 더 다양한 이미지 처리 방식을 학습하고 싶습니다. 아래는 매개변수로 'RGB'를 준 결과입니다.

```
from PIL import Image
import numpy as np
im = Image.open('chipmunk.png')
im = im.convert('RGB')
im2 = im.crop((280,150,430,300))
im2.save('chipmunk_head_RGB.png','PNG')
'chipmunk_head_RGB.png'
```

