HW 01 – REPORT

소속 : 정보컴퓨터공학부

학번 : 201924546

이름 : 이창민

1. 서론

이번 과제의 목표는 컴퓨터 비전을 위한 파이썬의 PIL, numpy 라이브러리의 아주 기본적인 내용들을 다뤄보는 것이다. 이미지를 열고, 이미지의 정보를 접근하고, 이미지를 모니터에 띄우고, 특정 부분을 자르고, 직접 각 pixel마다 연산을 적용하여 밝기를 조절하는 등의 실습을 진행한다.

2. 본론

먼저 주어진 이미지를 im.convert(‘L’)를 통해 greyscale로 바꾸어줄 수 있다. 이를 통해 나온 결과물은 다음과 같다.



이를 crop 함수를 통해 원하는 부분만 잘라낼 수 있는데, 다람쥐의 머리 부분만 떼어내면 다음과 같다.



또한 이 이미지는 현재 greyscale이기 때문에 결국 0~255 사이의 정수를 각 원소로 가진 2차원 배열이고, np.asarray(img)를 통해 실제 배열 객체로 만들어 줄 수 있다. np.mean(img) 함수를 사용하면 이미지의 평균 세기(밝기)를 쉽게 구하는 것도 가능하다.

np.asarray(img)를 통해 생성된 배열의 각 원소들은 수가 클수록 흰색에 가깝다. 이는 결국 각 원소들은 해당 픽셀의 밝기를 나타내는 것을 의미하고, 이를 이용해 사진의 전체적인 밝기를 조절할 수 있다. 밝기를 키우기 위해서는 각 원소를 더 커지도록 하면 된다. 따라서 각 원소에 특정 값을 더해주면 더 밝은 이미지가 생성될 것이다. 여기서 주의할 점은 각 원소의 최댓값은 255이므로 min(255, x)을 통한 컨트롤이 필요하다. 여기서 배열을 다시 이미지로 만드는 것은 Image.fromarray를 사용하면 되고, 결과물은 다음과 같다..



이번엔 이미지를 어둡게 하려고 한다. 원래 이미지에 특정 값을 빼도 되지만 이번엔 나눗셈을 사용한다. Array의 전체에 0.5를 곱해주면 절반의 세기를 갖게 될 것이다. 이미지의 각 픽셀의 원소는 정수형이어야 하므로 array를 정수형으로 바꿔준 뒤 img로 변환해주어야 한다. 이는 array.astype(‘uint8’)을 통해 구현이 가능하고, 결과물은 다음과 같다.



마지막으로 배열을 이용하여 그라데이션을 표현하는 우리만의 이미지를 직접 생성해보자. 이미지의 왼쪽 부분은 어둡다가, 오른쪽으로 갈수록 점점 밝아지는 그라데이션을 표현하고 싶다면

[0, 1, 2, . . . , 253, 254, 255] 이라는 1차원 배열을 256개 넣으면 된다. 이 1차원 배열 하나를 생성하는 것은 np.arange(0, 256)을 통해 구현이 가능하고, 이 1차원 배열을 256번 반복하여 넣는 것은 np.tile(array, [256, 1])을 통해 구현이 가능하다. 이를 통해 생성한 이미지는 다음과 같다.



3. 결론

이렇게 파이썬을 통해서 이미지를 처리하는 것을 실습했다. 이미지의 원하는 부분을 잘라내고, 밝기를 높이고, 밝기를 낮추는 등 이론적으로 배운 부분을 결과물로 확인하면서 잘 동작하는 것을 확인할 수 있었다. 실제 이미지의 프로세싱은 배열을 다루는 것으로 진행되기 때문에 numpy배열에 대한 숙련도가 필요해 보인다. 또한 마지막에 생성한 그라데이션 이미지와 실제 이미지를 일대일로 곱하여 [0, 255]의 값으로 변환해준다면 실제 이미지에 그라데이션 효과를 넣을 수도 있을 것 같아서 시도해보았더니 다음과 같은 사진 또한 얻을 수 있었다.

