

Introduction to Computer Vision HW01 – Eagle’s Head

2021-1학기 컴퓨터비전개론 059분반 2017244337 김재현

주어진 vision_hw1.py 파일은 PIL, numpy 패키지를 이용하여 원본 파일에서 3개의 편집된 이미지를 생성하고, 1개의 새로운 이미지를 만들어 내는 코드입니다. 각각의 이미지가 생성되는 과정은 아래와 같습니다.

첫 번째 이미지인 'eagle_head.png'를 생성하기 위해 우선 원본 이미지를 im이라는 변수에 할당합니다. 그 후 Image 클래스의 convert 함수를 이용하여 이미지를 그레이스케일로 바꾸어 줍니다. 이 이미지를 가로 450~550픽셀, 세로 300~400픽셀의 범위만큼 잘라내어, im2라는 변수에 할당합니다. 이 이미지를 save 함수를 통해 PNG 형식으로 저장하면 'eagle_head.png' 파일이 생성됩니다.



그림 1) eagle_head.png

두 번째 이미지는 첫 번째 이미지인 im2에서 각 픽셀의 값을 50만큼 증가시킨 것입니다. 이 연산을 위해 numpy의 add 함수를 이용하여 해당 Image 클래스의 이미지를 numpy 배열로 변환합니다. 이후 이 배열을 im3_array라는 새로운 변수에 복사하여 할당한 뒤, im3_array의 각 원소의 값을 50씩 증가시킵니다. 이때 각 원소, 즉 픽셀의 intensity는 최대 255이므로, 계산된 값이 255를 넘지 않도록 min 함수를 이용하여 값을 clipping 해줍니다. 이렇게 계산된 배열을 Image 클래스의 fromarray 함수를 이용하여 변환한 후 저장하면 아래와 같은 'eagle_head_bright.png' 파일이 생성됩니다.



그림 2) eagle_head_bright.png

세 번째 이미지는 두 번째 이미지와 반대로 기존 im2의 각 픽셀의 값을 50% 감소시킨 것입니다. 이 연산을 위해 im2_array를 복사 후 할당하고, 이렇게 할당된 변수 im4_array 원소 전체에 0.5를 곱하여 밝기를 감소시킵니다. 다만 이 연산 이후에 각 원소가 float 형으로 변하기 때문에, astype 함수를 통해 배열 im4_array를 uint8 형으로 변환해야 합니다. 이 배열을 마찬가지로 fromarray 함수를 이용하여 이미지로 변환한 뒤 저장하면, 아래와 같은 'eagle_head_dark.png' 파일이 생성됩니다.



그림 3) eagle_head_dark.png

마지막 4번째 이미지는 numpy를 통해 배열 값을 직접 생성하여 만들어 낸 그라데이션 이미지입니다. 우선 numpy의 `arrange` 함수를 이용하여 0부터 255까지의 값을 가지는 1차원 배열을 생성하고, 이를 `grad` 변수에 할당합니다. 이후 `tile` 함수를 이용하여 이 배열을 256개의 행에 차례로 나열한 2차원 배열을 생성하고, 이를 다시 `grad` 변수에 할당합니다. 이렇게 생성된 배열을 `fromarray` 함수를 이용하여 이미지로 변환한 뒤 저장하면, 아래와 같은 'gradient.png' 파일이 생성됩니다.



그림 4) gradient.png

이처럼 PIL 패키지의 이미지 관련 함수와 numpy 패키지의 배열 관련 함수를 이용하여 기존의 이미지를 수정하고, 새로운 이미지를 생성할 수 있었습니다. 다만 위에서 설명한 예제 코드를 실행시키기 위해서는 사전에 Python 패키지 매니저인 pip를 통해 pillow 패키지와 numpy 패키지를 설치하는 과정이 필요하였습니다.