

구현 내용

이번 과제는 opencv 라이브러리를 이용해 불러온 이미지 파일을 rgb에서 gray 로 바꾼 뒤 저장 할 수 있도록 행렬을 만드는 것이다. 핵심은 opencv로 이미지 파일을 불러올 때 행렬 형태로 불러오기 때문에 임의로 조정이 가능하다는 것이다. 이 행렬을 numpy 라이브러리를 이용해 변형시킨다.

이유(구현 방법)

우선 처음엔 이미지를 불러올 때 어떤 행렬의 형태로 불러오는 것을 아는 것인가가 중요했다. 그래서 확인해 봤는데 (x좌표, y좌표, rgb값) 의 형태로 불러오는 것을 알아냈다. 또 gray의 형태는 (x좌표, y좌표) 가 된다는 것을 알았다. 예를 들어 내가 다운받은 예시의 그림을 color 형태로 불러오면 (956,1300,3), gray 형태로 불러오면 (956,1300) 형태로 불러오며, color 행렬은 $[[[71, 52, 50], \dots]]$, gray 행렬은 $[[55],[55],\dots]$ 가 된다. 그리고 이론시간에 배운게 rgb엔 가중치를 각각 0.2125, 0.7154, 0.0721를 곱해주니까 이걸 더해서 하나의 원소가 나온 결과가 gray의 (956,1300) 형태가 된다는 것을 알아냈다.

그래서 우선 처음엔 color로 불러와 (956,1300,3) 형태가 된다. (956,1300) 차원의 행렬을 만들어 (.,3) 번째의 원소를 가중치에 맞게 더해 해당 index에 결과를 넣는다. 이것을 이중 for 문을 통해 넣었었다. 작동은 잘 되었으나 실행하는데 너무 시간이 오래걸리고, 내장함수인 cvtColor를 통해 gray로 변형했을 때는 적은 시간만 걸리는 것이다. 그래서 더 효율적인 계산 방법이 있을 거라고 생각해 행렬 간의 곱셈 관계를 살펴보던 중, 차원을 줄이는데 줄이는 과정에서 각 원소의 가중치를 더해 줄일 수 있는 함수가 있다는 것을 알아냈다.

그 함수는 numpy의 average 함수인데, axis와 weights 매개변수를 통해 내가 원하는 가중치 만큼 더할 수 있다는 것을 알았다. 그래서 해봤는데 잘 작동하고 작동 시간도 현저히 줄어들었다.

결과화면을 밑에 첨부한다.

```

1 import cv2
2 import numpy as np
3 import matplotlib.pyplot as plt
4
5 def my_bgr2gray(bgr_array):
6     # new_array = np.ones((len(bgr_array), len(bgr_array[0])))
7     #
8     # for row in range(0, len(bgr_array)):
9     #     for col in range(0, len(bgr_array[0])):
10     #         new_array[row][col] = np.around(np.dot([0.2125, 0.7154, 0.0721], bgr_array[row][col]))
11     #
12     # return new_array
13     return np.round(np.average(colorscale, axis=2, weights=[0.2125, 0.7154, 0.0721]))
14
15 colorscale = cv2.imread('./image.jpg', cv2.IMREAD_COLOR)
16 print(colorscale)
17 print(colorscale.shape)
18 f_g = my_bgr2gray(colorscale)
19 plt.imshow(f_g)
20 cv2.imwrite('./image_custom_f.jpg', f_g)

```

```

Run: my_bgr2gray x
...
[ 72  51  50]
[ 72  51  50]
[ 72  51  50]]
(956, 1300, 3)
[[37. 36. 34. ... 45. 42. 39.]
 [26. 29. 35. ... 31. 30. 29.]
 [34. 30. 27. ... 26. 25. 25.]
 ...
 [55. 55. 55. ... 55. 55. 55.]
 [55. 55. 55. ... 55. 55. 55.]
 [55. 55. 55. ... 55. 55. 55.]

Process finished with exit code 0

```

느낀 점

opencv2에서 이미지를 행렬 형태로 어떻게 저장하는지를 이해하는데 처음 접한 행렬의 형태라 이해하는데 조금 어려웠다.

과제 난이도

어렵다. 이런 함수가 있지 않을까 생각해서 찾는게 좀 복잡했음