

디지털영상처리

과제1 – 트랙바 제어로 영상 밝게 만들기

2021. 11. 04. 목

컴퓨터공학과

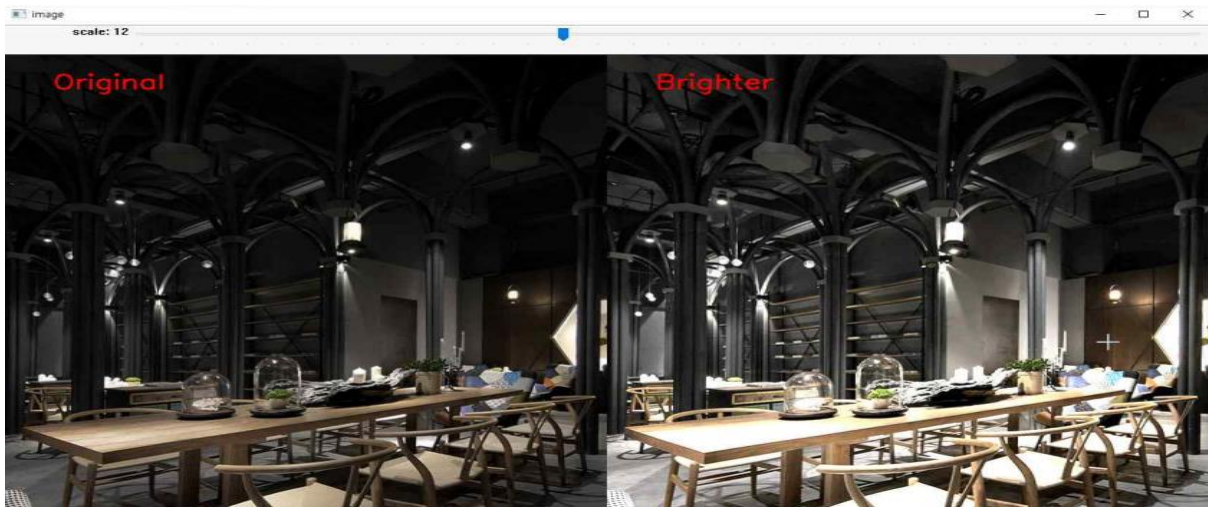
2019305059

이현수

목차

1. 문제 정의
2. 문제풀이 절차
3. 소스코드
4. 실행결과
5. 시행착오, 배운점, 기술적의의

1. 문제 정의



원하는 영상을 읽어 들여 이 영상의 밝기를 트랙 바로 제어하는 프로그램이다. 이때 원본영상과 밝게 만든 영상을 나란히 배열하여 출력한다. 좌측에는 원본영상을 우측에는 밝게 만든 영상을 출력한다. 좌측의 원본영상에는 빨간색 텍스트로 'Original'을 출력하고, 우측의 밝게 만든 영상에는 'Brighter'의 빨간색 텍스트를 출력한다.

트랙 바는 0~30의 값을 가진다. 이는 원본 영상의 0%, 10%, 20%, 30%, ... 290%, 300% 만큼 더 밝게 만들라는 의미이다. 따라서 트랙바가 0을 가리키면 원본영상과 같다. 이 처리는 원본 영상에 대한 곱셈 배수를 곱해서 구현한다.



만약 key 's'를 누르면, 밝아진 영상을 현재 폴더에 지정된 파일 이름으로 저장한다. 사진을 저장할 때는 'Brighter' 문자정보를 저장 영상에 포함하지 않는다. 파일 이름은 'tmpxx_파일이름.png' 형식이다.

만약 key 'esc'를 누르면 프로그램이 종료된다.

2. 문제풀이 절차

1) 파일 읽기

문제의 조건에 나온것처럼 소스파일 상단에 Path과 Name을 지정한다. cv2의 imread 함수를 통해 이미지를 읽어온다. 이때 이미지를 읽어오면서 나누기 255를 통해 0~1로 정규화 시키고 타입은 부동소수가 된다.

2) 원본이미지 복사하기

원본이미지를 .copy해서 복사본을 만든다. 이유는 원본 영상에 'Original' 텍스트를 출력시켜 나타내야 하기 때문이다. 이때 텍스트가 출력되지 않는 원본영상은 밝게 만들 때 소스 이미지로 사용해야 하기 때문에 'Original' 텍스트를 출력시키기 위한 복사 이미지를 만들어 이 이미지를 가지고 계속 프로그램 왼쪽 원본 영상에 나타낸다.

3) 트랙바 만들기

Cv2 모듈에 내장된 트랙바를 통해 트랙바를 만든다. 문제 조건에서는 scale을 조정하는 것만 있다. 조건은 0~30까지이다.

4) 트랙바 이벤트 함수 만들기

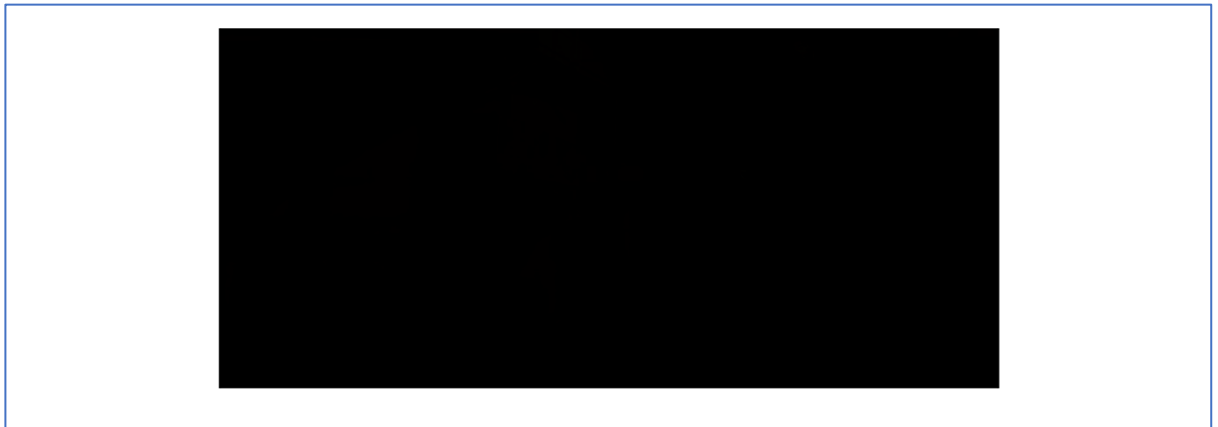
트랙바에서 scale을 조정하면 호출되는 이벤트 함수를 만들어 등록한다. 여기서는 callback_scale 메소드를 이벤트 함수로 만들어 등록했다. Callback_scale 메소드는 전역변수 scale과 Touched에 대해 scale은 사용자가 설정한 값으로 초기화 시키고, Touched는 False에서 True로 만들어준다. 이렇게 되면 while 무한 반복문에서 if Touched==True 조건이 만족되어 영상을 밝게 만들어주는 소스코드가 실행된다.

5) 사진 밝게 만들기

원본이미지를 복사한다. 복사한 이미지를 가지고 사진을 밝게 만들어 출력시킨다. 복사한 이미지에 $(1 + 0.1 \cdot \text{scale})$ 을 곱해주면 scale값이 1이면 원본영상의 10%만큼 밝게되고, scale 값이 30이면 300% 밝게된다. 그 후 'Brighter' 텍스트를 밝게만든 사진에 출력시키고 hconcat 메소드를 통해 원본영상과 밝게 만든 영상을 가로로 붙여 출력시킨다.

6) Key 입력값 's'가 들어올 때 구현

키 's'가 입력되면 현재 트랙바 scale로 설정되어 밝게된 영상을 텍스트를 제외한 이미지를 tmpxx_파일이름.png로 저장한다. 우선 원본이미지를 복사한다. 그 후 현재 트랙바 scale에 설정된 크기만큼 밝게 만든다. 여기서 현재 원본이미지는 0~1로 정규화된 부동소수 이미지이다. 처음 이미지를 읽어올 때 나누기 255를 했기 때문이다.



여기서 그냥 imwrite 메소드를 통해 저장시키면 위 사진처럼 검정화면이 나온다. 그래서 밝게 만든 이미지에 곱하기 255를 통해 범위를 다시 0~255로 만들어야 정상적인 사진이 저장된다.

7) Key 입력값 'esc'가 들어올 때 구현

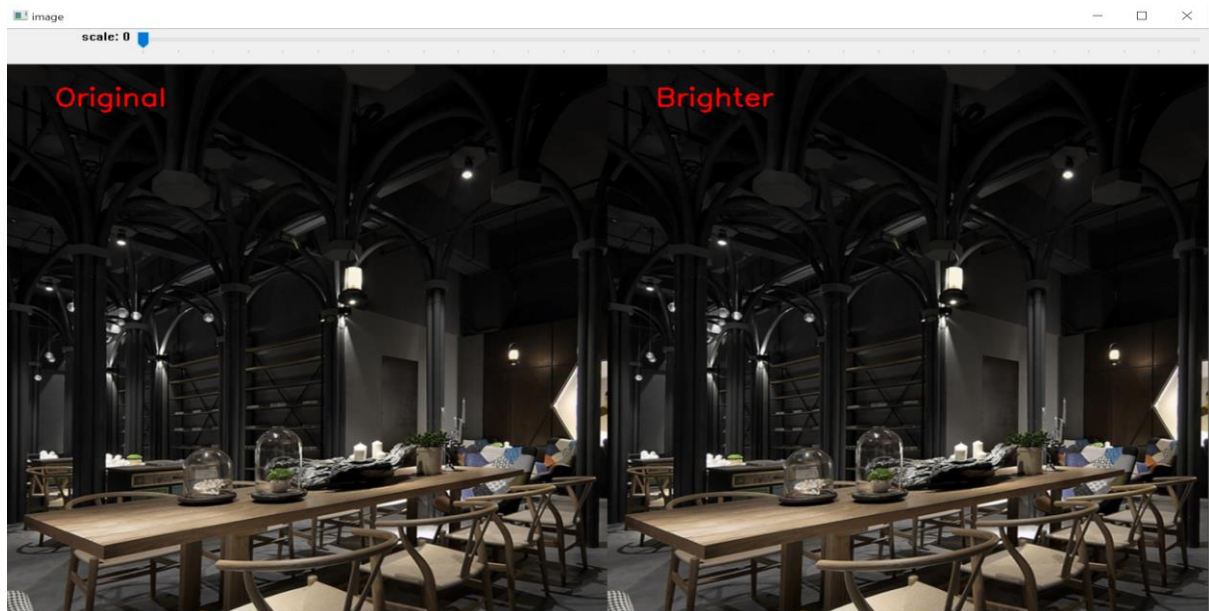
키 'esc'를 누르면 프로그램을 종료시켜야 된다. esc키는 아스키코드 10진수로 27이므로 $k==27$ 일경우 조건문을 만들어 while 무한 반복문에서 break 시켰다. 그 후 cv.destroyAllWindows 메소드를 통해 모든창을 닫았다.

3. 소스코드

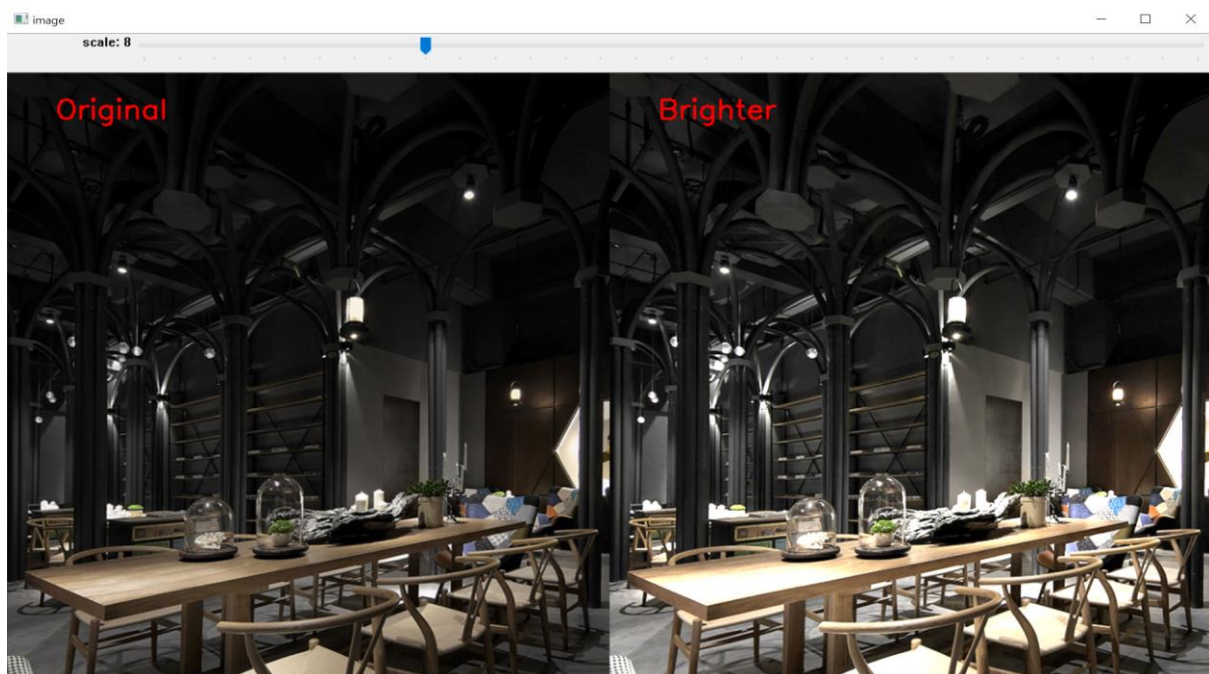
```
1 # 아래 Path와 Name은 그대로 현재 설정을 사용해 주세요....
2 # 현재 d:/dip/dark1.png 파일을 입력 영상으로 사용하고 있습니다.
3 Path = 'd:/dip/'
4 Name = 'dark1.png'
5
6 import cv2 as cv
7
8 FullName = Path + Name
9 image_ori = cv.imread(FullName)/255 # 이미지 파일 읽어오기 + 0~1로 정규화
10
11 # 원본 이미지 복사해 'Original' text 그리기
12 image_ori_text = image_ori.copy()
13 cv.putText(image_ori_text, 'Original', (50, 50), cv.FONT_HERSHEY_SIMPLEX, 1, (0, 0, 255), 2)
14
15
16 Touched = False # 트랙바를 건드리면 True. 이 때만 연산을 다시 한다.
17 scale = 0 # 밝기 scale 크기
18
19 # scale 이벤트 함수
20 def callback_scale(x):
21     global scale, Touched
22     scale = x
23     Touched = True
24
25 cv.namedWindow('image')
26 cv.createTrackbar('scale', 'image', 0, 30, callback_scale)
27
28 image_scale = image_ori.copy() # 원본 이미지 복사
29 cv.putText(image_scale, 'Brighter', (50, 50), cv.FONT_HERSHEY_SIMPLEX, 1, (0, 0, 255), 2) # 복사이미지에 'Brighter 출력'
30 image_h = cv.hconcat([image_ori_text, image_scale]) # 원본이미지, 복사이미지 가로로 붙이기
31 cv.imshow('image', image_h) # 이미지 출력
32 while(1):
33     k = cv.waitKey(1)
34     if k == 115: # 's' 키를 누르면 사진 저장
35         image_scale_save = image_ori.copy() # 원본영상 복사
36         image_scale_save = image_scale_save * (1+0.1*scale) # 밝게 만들기
37         image_scale_save = image_scale_save * 255 # 0~255로 정규화
38         image_save_filename = f'tmp{scale:0>2}_{Name[:Name.rfind(".")].png}' # 저장할 파일 이름
39         cv.imwrite(image_save_filename, image_scale_save) # 이미지 저장
40     elif k== 27: # esc 키를 누르면 종료
41         break
42     if Touched==True:
43         image_scale = image_ori.copy() # 원본영상 복사
44         image_scale = image_scale * (1 + 0.1 * scale) # 밝게 만들기
45         cv.putText(image_scale, 'Brighter', (50, 50), cv.FONT_HERSHEY_SIMPLEX, 1, (0, 0, 255), 2) # 'Brighter 출력'
46         image_h = cv.hconcat([image_ori_text, image_scale]) # 원본이미지, 밝게만든 이미지 가로로 붙이기
47         cv.imshow('image', image_h) # 이미지 출력
48         Touched=False
49
50 cv.destroyAllWindows()
```

4. 실행결과

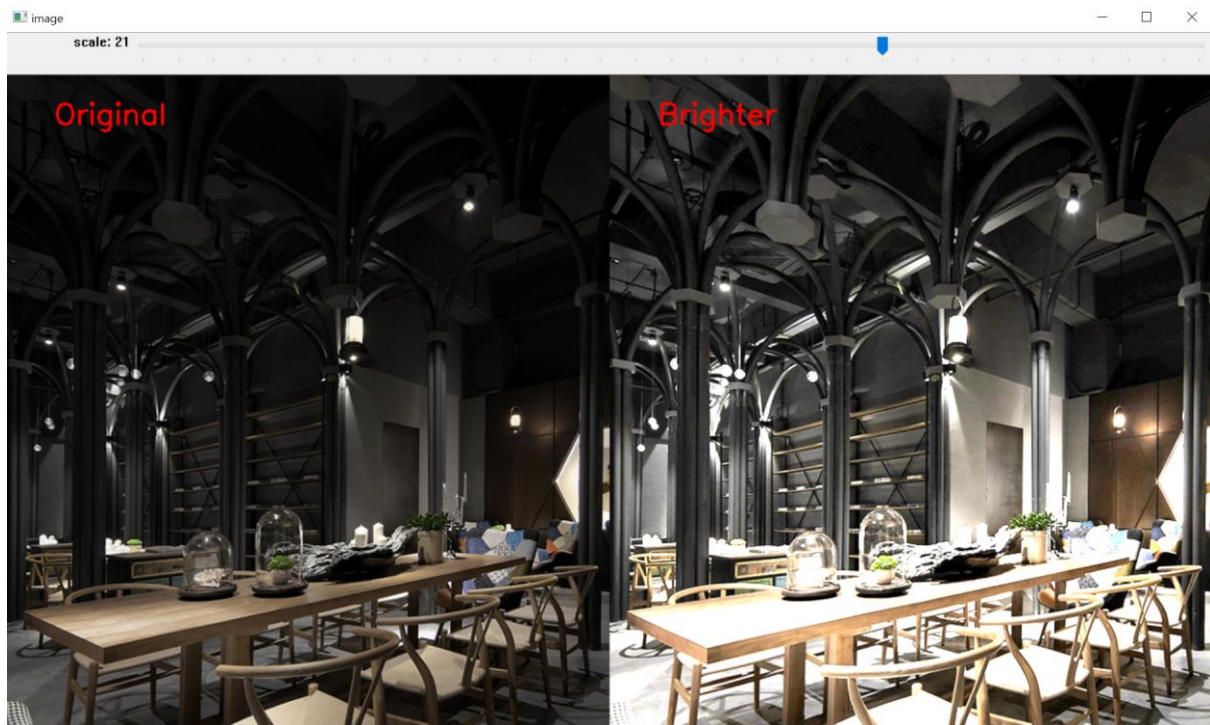
1) 테스트1



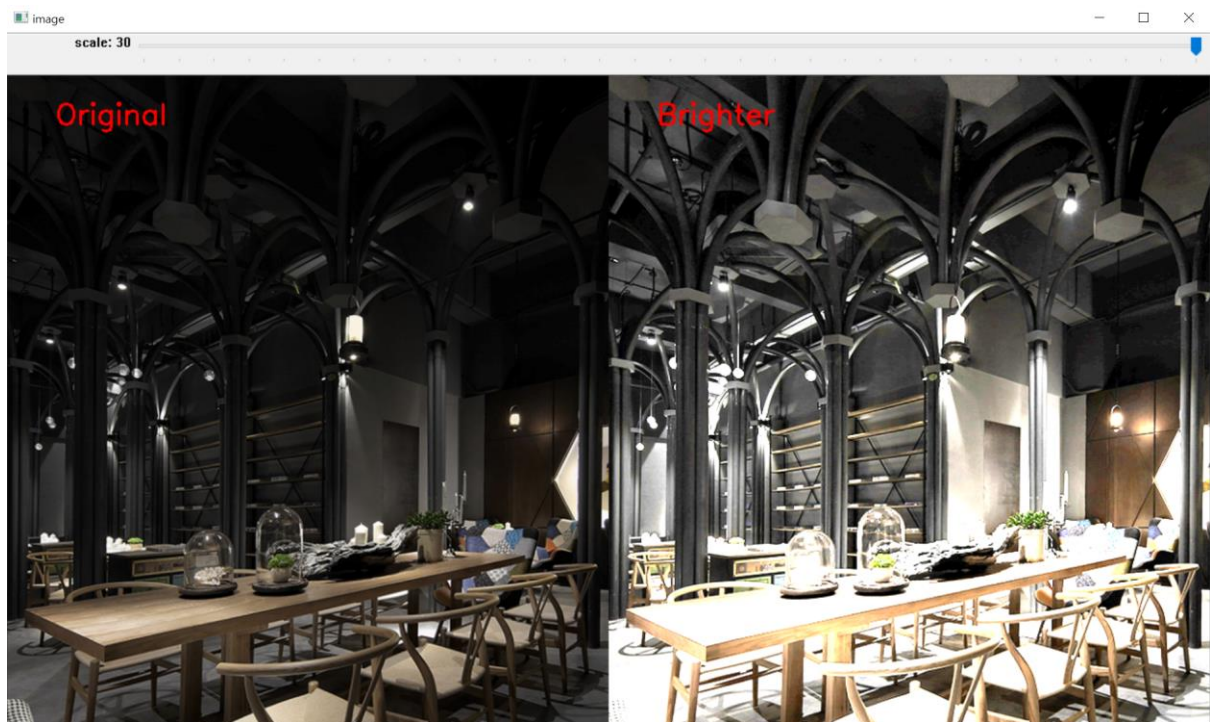
처음 실행 화면. Scale이 0이라 양쪽 이미지가 같다.



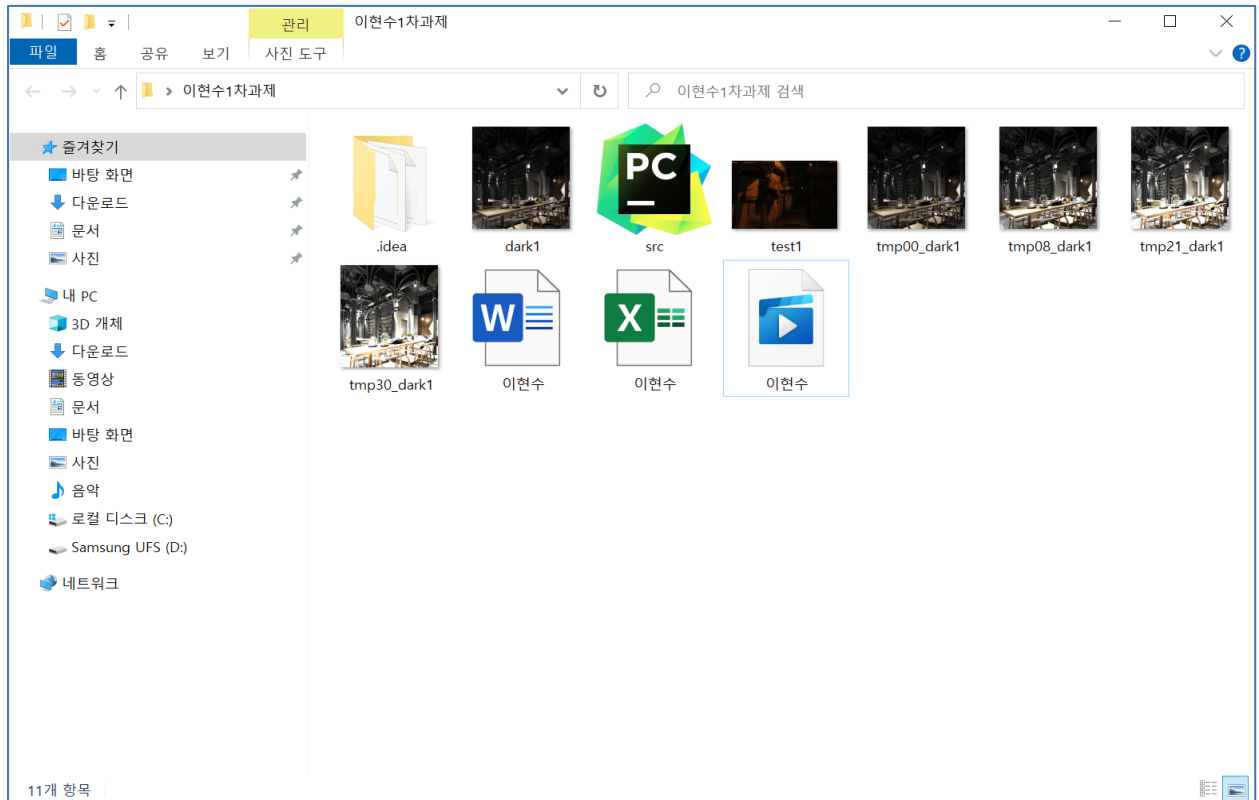
Scale = 8인 경우



Scale = 21인 경우

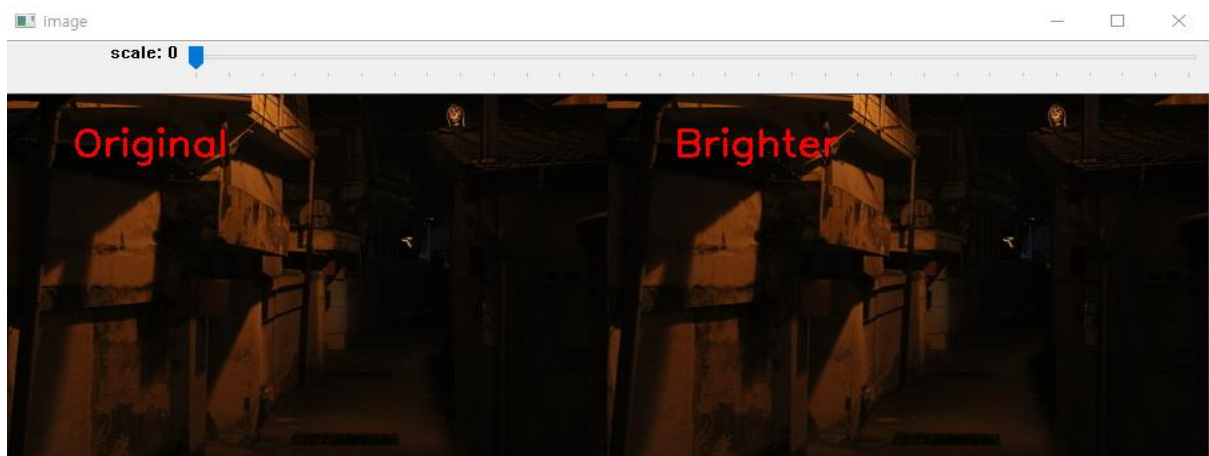


Scale = 30인 경우

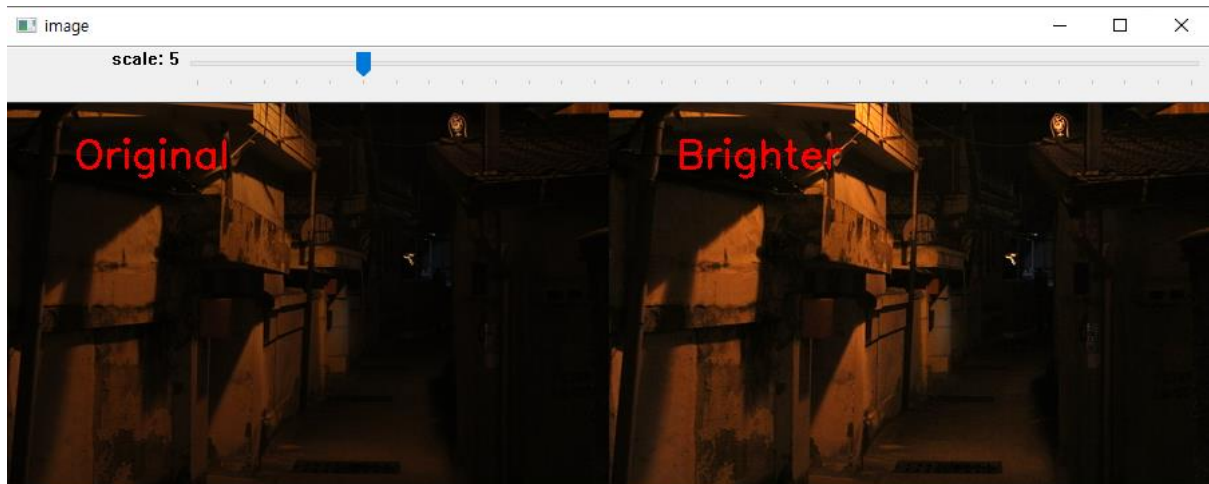


Scale이 0, 8, 21, 30일 때 's'키를 눌렀다. 소스코드가 있는 폴더에 이미지가 원하는 파일 이름으로 'Brighter' 텍스트가 지워진 이미지로 정상적으로 저장된 것을 확인할 수 있다.

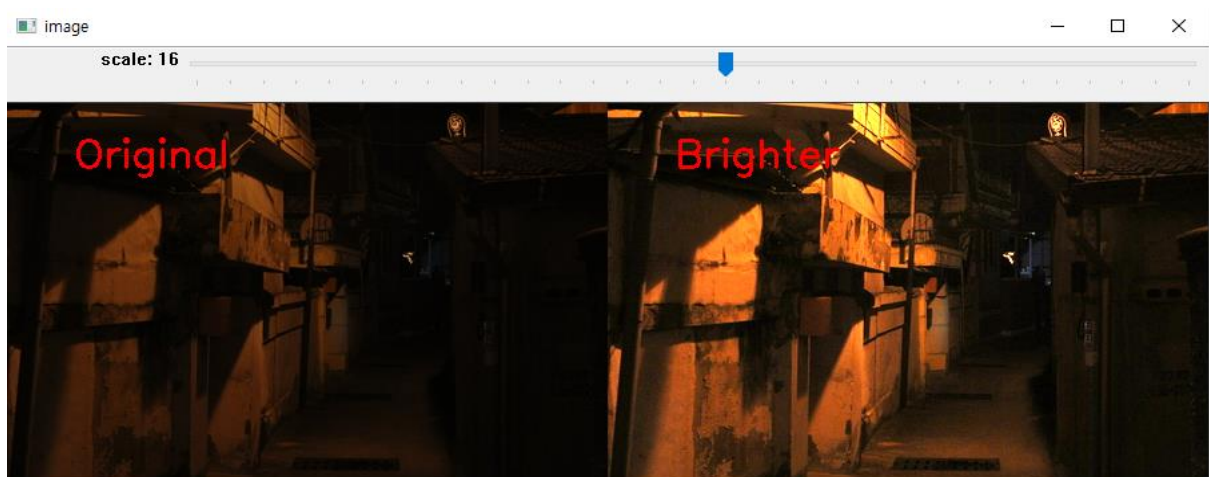
2) 테스트2



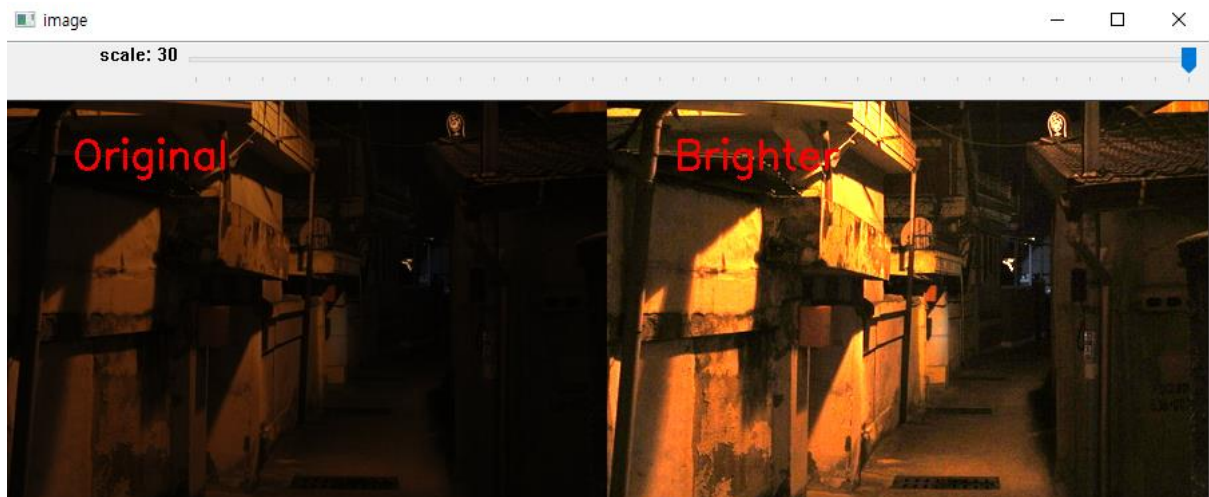
Scale = 0 인 경우, 원본 영상과 같다.



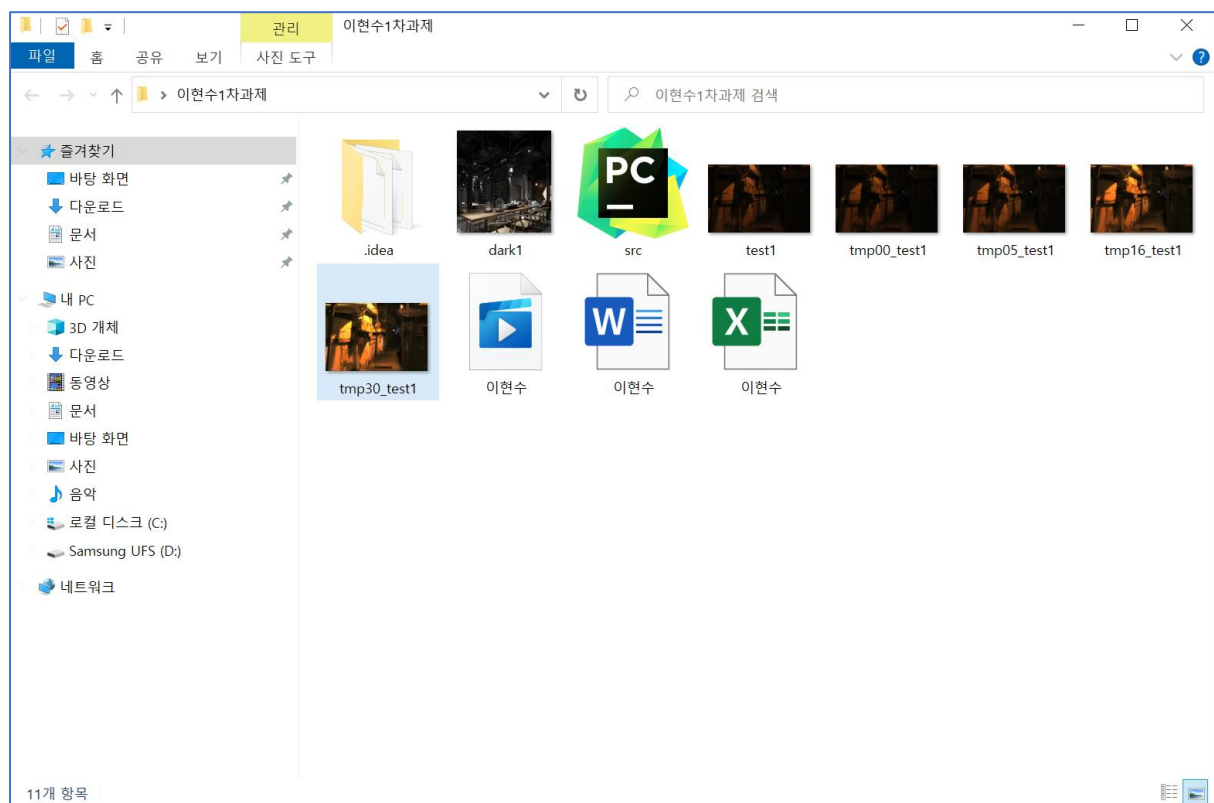
Scale = 5 인 경우



Scale = 16 인 경우



Scale = 0 인 경우



Scale이 0, 5, 16, 30일 때 's'키를 눌렀다. 소스코드가 있는 폴더에 이미지가 원하는 파일 이름으로 'Brighter' 텍스트가 지워진 이미지로 정상적으로 저장된 것을 확인할 수 있다.

5. 시행착오, 배운점, 느낀점, 기술적의의

1) 시행착오 1 - 같은 타입의 이미지끼리만 hconcat이 됨.

처음에 imread를 통해 이미지를 받아오면 0~255범위의 uint8 자료형이다. 사진을 밝게 하기 위해서 0~1로 정규화 하고 곱을 하면 부동소수 자료형이 된다. 이때 원본영상의 uint8과 밝게한 사진의 부동소수 자료형 사진 두개를 hconcat 했지만 오류가 났다. hconcat을 하기 위해서는 아마 두 개의 이미지 영상의 자료형이 같아야 하는 것 같다.

이것을 해결하기 위해 원본이미지를 읽어오면서 나누기 255를해 0~1로 정규화 시키고 부동소수 자료형 이미지로 만들어 해결했다.

2) 시행착오 2 – imwrite

사진을 밝게 만든 이미지 영상은 0~1로 정규화된 부동소수 자료형이다. 여기서 imwrite를 해보니 검정색 이미지만 저장이 되었다. 그래서 밝게 만든 영상에 255를 곱하니 정상적으로 이미지 영상이 저장되었다.

3) 배운점 및 느낀점

이미지 영상을 밝게 만들기 위해서는 감마 변환 등 여러가지가 있지만 직접 특정 값을 곱해서 이미지를 밝게하는 것을 알게 되었다.

시행착오1, 2에서 겪은 같은 타입끼리만 hconcat이 되고, imwrite는 0~1로 정규화된 이미지를 정상적으로 저장하지 못한다는 사실을 배웠다.

트랙바를 사용한 프로그램은 처음 만들어봐서 좋은 경험이었고, 어두운 사진을 영상처리를 통해 밝게 만드는데 신기했고, 이 프로그램은 과제뿐만 아니라 일상 생활에서 어두운 사진을 밝게 만들고 싶을 때 실제로 이용할거 같아서 좋았다.

4) 기술적 의의

이미지를 밝게 만들 때 opencv에서 제공하는 여러 함수를 사용하지 않고 직접 계산해 이미지를 밝게 만들.