

```

1
2 # 기본적 영상 입출력
3 #
4 # 실행 방법 (아나콘다 설치된 컴퓨터에서 CMD 명령어 창에서 아래와 같이 실행)
5 # conda create -n ip opencv          ip 이름을 갖는 가상환경 생성
6 # conda activate ip                  ip 가상환경 활성화
7 # python image_basic.py              프로그램 실행
8
9 import cv2
10 import sys, numpy
11
12 #=====
13 # [1] OpenCV형 img 객체로 전달받은 영상데이터를
14 #     windowName이름의 창(window)을 만든 후, 그 안에 img의 내용을 출력
15 #=====
16 def show_image_window(img, windowName='IMAGE', position=[0,0]) :
17     cv2.namedWindow(windowName,cv2.WINDOW_AUTOSIZE)          # 새로운 창 생성
18     cv2.moveWindow(windowName, position[0], position[1])      # x:position[0], y 위치로 창 이동
19     cv2.imshow(windowName, img)                                # 화면에 창 출력
20     # end of the fn.
21
22 #=====
23 # [1] 영상 파일 입출력과 기본 처리
24 #=====
25 if __name__ == '__main__' :                                  # 직접 실행한 경우이면
26     while True:
27         print('파일 크기가 크면 실행시 시간이 걸릴 수 있습니다. [강제종료는 컨트롤키와 c키를 동시에 누르세요]')
28         print('파일명에 공백문자가 안들어가도록 하고, 파일확장자도 반드시 붙이세요\n')
29         fileName = input('영상파일명 입력:')                  # 그림 파일명 입력
30         print('입력한 파일명은 [{}] 입니다'.format(fileName))
31         #fileName = 'sample1.png'
32
33         img = cv2.imread(fileName,cv2.IMREAD_UNCHANGED)      # 영상데이터를 읽어 img 객체에 저장
34
35         if img is None:                                       # 제대로 불러오지 못했으면
36             sys.exit("영상 파일 열기 오류")                  # 프로그램 종료
37
38         # [1] 이미지와 이미지 정보 출력
39         nameString = fileName + str(img.shape)                # 창이름을 파일명과 img구조로 문자열로 설정
40         show_image_window(img, windowName=nameString, position=[0,0]) # 출력을 위해 위에서 만든 출력용 함수 호출
41
42         # [2] 영상 채널별 분리
43         heightSize, widthSize, channelSize = img.shape        # 그림의 세로, 가로 픽셀수와 채널 수
44         print('입력영상 정보: 세로 {}픽셀, 가로{}픽셀, {}개 채널을 갖는 영상입니다'.
45               format(heightSize, widthSize, channelSize))
46         if channelSize == 4:
47             img = cv2.cvtColor(img, cv2.COLOR_BGRA2BGR)       # 4채널 -> 3채널 영상으로 포맷 변경
48             print("Alpha Channel Removed")
49         elif channelSize < 3:
50             img = cv2.cvtColor(img, cv2.COLOR_GRAY2BGR)       # 1채널 -> 3채널 영상으로 포맷 변경
51             print("Gray changed to Color")
52
53         imgBlueChnl, imgGreenChnl, imgRedChnl = cv2.split(img) # 각 채널별로 픽셀별 값 정보(0~255) 출력
54
55         show_image_window(imgRedChnl, windowName='RED'+str(imgRedChnl.shape), position=[100,10])
56         show_image_window(imgBlueChnl, windowName='GREEN'+str(imgBlueChnl.shape), position=[100,50])
57         show_image_window(imgGreenChnl, windowName='BLUE'+str(imgGreenChnl.shape), position=[100,100])
58
59         # [3] 채널 합치기
60         img2 = cv2.merge((imgRedChnl, imgGreenChnl, imgBlueChnl)) # 새 채널의 색상을 채널 위치를 바꾸어서 합치기
61         show_image_window(img2, windowName='CHANNEL_SUM img2'+str(img2.shape), position=[300,500])
62
63         # [4] 산술 연산 (두 영상의 shape이 같아야 함)
64         img3 = 255 - img                                         # img3 영상에 대하여 보색 구하기
65         show_image_window(img3, windowName='COMPLEMENTARY img3'+str(img3.shape), position=[500,500])
66
67         img4 = img3 // 2                                         # img4 영상에 대하여 밝기를 1/2로 줄이기
68         show_image_window(img4, windowName='DIVIDED img4'+str(img4.shape), position=[700,500])
69
70         img5 = cv2.add(img3, img4)                               # 두 개의 영상(img3, img4) 더하기 (0보다 작으면 0으로, 255보다 크면 255로 제한)
71         show_image_window(img5, windowName='ADD img5'+str(img5.shape), position=[900,500])
72
73         img6 = cv2.addWeighted(img3, 0.7, img4, 0.3, 0)        # 두 개의 영상(img3, img4) 가중치(70%, 30%)별로 더하기

```

```

74 show_image_window(img6, windowName='ADD_WEIGHTED img6'+str(img6.shape), position=[900,500])
75
76 img7 = img3 + img4 # 영상 더하기 (나머지 연산, 250 + 10 = 260 => 260 % 256 = 4)
77 show_image_window(img7, windowName='Add by Residue img7'+str(img7.shape), position=[900,500])
78
79 img8 = cv2.bitwise_and(img3, img4) # 영상 비트단위로 AND 하기
80 show_image_window(img6, windowName='Bitwise AND img8'+str(img8.shape), position=[900,500])
81
82 # [5] 화소별 접근 및 연산 (예:빨간색, 파란색 강조)
83 img9 = img.copy() # img와 같은 영상 img9 만들기
84 img10 = img.copy() # img와 같은 영상 img10 만들기
85
86 # 모든 점에 대하여 색상값을 지정 하기
87 for row in range(heightSize): # 가로로 먼저 loop를 돌리고
88     for col in range(widthSize):
89
90         b, g, r = img[row, col] # 원본 img의 row번째줄, col번째 위치 칼라 픽셀값(0~255) 가져오기
91
92         if r > 150 and b < 200 and g < 200 : # 빨간색 값이 충분하고 흰색이 아니면 빨간색 채널값 255로
93             img9[row, col] = (0, 0, 255)
94         elif b > 150 and r < 200 and g < 200 : # 파란색 값이 충분하고 흰색이 아니면 파란색 채널값 255로
95             img10[row, col] = (255, 0, 0)
96
97 show_image_window(img9, windowName='RED COLOR CHECK img9'+str(img9.shape), position=[900,500])
98 show_image_window(img10, windowName='BLUE COLOR CHECK img10'+str(img10.shape), position=[1100,500])
99
100 print('\n그림창 아무거나 마우스로 선택한 상태에서 아래 버튼을 누르세요')
101 print('[s]- 영상 저장, [q]- 프로그램 종료, [다른 키]- 다른 영상으로 계속 실행')
102 keyInChar = cv2.waitKey(0) # 키보드 입력 무한정 기다림
103
104 cv2.destroyAllWindows() # OPENCV를 이용해서 생성한 창들 모두 종료
105
106 if keyInChar == ord("s") : # 소문자 s키를 눌렀으면 실행
107     # 결과 그림 4장만 저장 해보기 (확장자명에 그림파일 포맷 지정해야 함)
108     cv2.imwrite("img2.jpg", img2) # img2 객체에 저장된 영상 데이터를 img2.jpg 이름으로 저장
109     cv2.imwrite("img3.jpg", img3)
110     cv2.imwrite("RED.jpg", img7)
111     cv2.imwrite("BLUE.jpg", img8)
112
113 elif keyInChar == ord("q") :
114     break
115

```