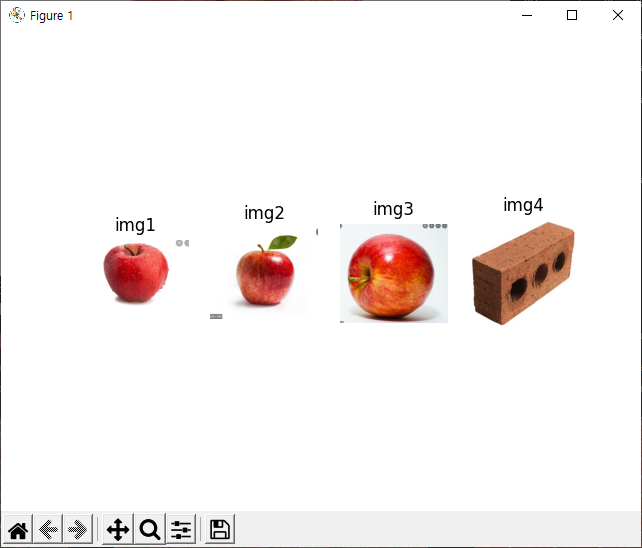
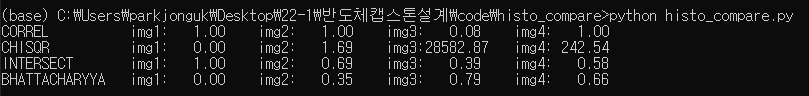
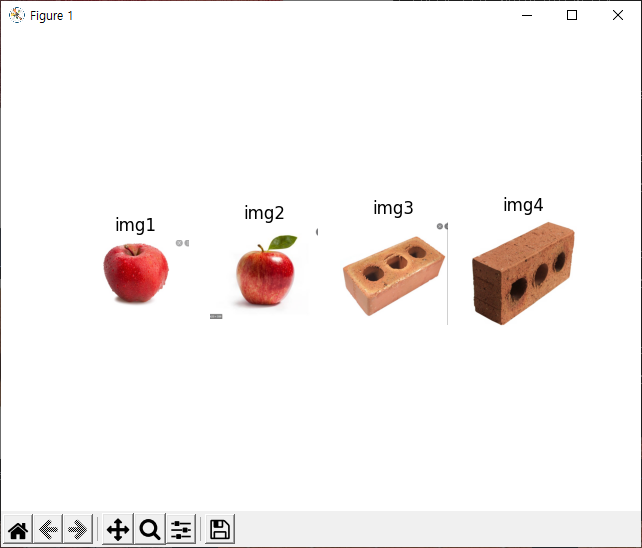
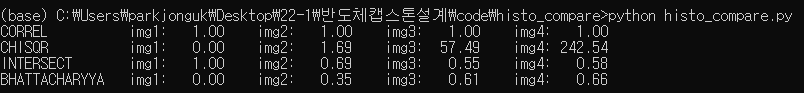


https://bkshin.tistory.com/entry/OpenCV-12-%EC%9D%B4%EB%AF%B8%EC%A7%80-%EC%9C%A0%EC%82%AC%EB%8F%84-%EB%B9%84%EA%B5%90-%EC%82%AC%EB%9E%8C-%EC%96%BC%EA%B5%B4%EA%B3%BC-%ED%95%B4%EA%B3%A8-%ED%95%A9%EC%84%B1-%EB%AA%A8%EC%85%98-%EA%B0%90%EC%A7%80-CCTV









# 히스토그램 비교 (histo\_compare.py)

import cv2, numpy as np

import matplotlib.pylab as plt

img1 = cv2.imread('../img/apple-1.png')

img2 = cv2.imread('../img/apple-2.png')

img3 = cv2.imread('../img/brick-2.png')

img4 = cv2.imread('../img/brick-1.png')

cv2.imshow('query', img1)

imgs = [img1, img2, img3, img4]

hists = []

for i, img in enumerate(imgs) :

    plt.subplot(1,len(imgs),i+1)

    plt.title('img%d'% (i+1))

    plt.axis('off')

    plt.imshow(img[:,:,::-1])

    #---① 각 이미지를 HSV로 변환

    hsv = cv2.cvtColor(img, cv2.COLOR\_BGR2HSV)

    #---② H,S 채널에 대한 히스토그램 계산

    hist = cv2.calcHist([hsv], [0,1], None, [180,256], [0,180,0, 256])

    #---③ 0~1로 정규화

    cv2.normalize(hist, hist, 0, 1, cv2.NORM\_MINMAX)

    hists.append(hist)

query = hists[0]

methods = {'CORREL' :cv2.HISTCMP\_CORREL, 'CHISQR':cv2.HISTCMP\_CHISQR,

           'INTERSECT':cv2.HISTCMP\_INTERSECT,

           'BHATTACHARYYA':cv2.HISTCMP\_BHATTACHARYYA}

for j, (name, flag) in enumerate(methods.items()):

    print('%-10s'%name, end='\t')

    for i, (hist, img) in enumerate(zip(hists, imgs)):

        #---④ 각 메서드에 따라 img1과 각 이미지의 히스토그램 비교

        ret = cv2.compareHist(query, hist, flag)

        if flag == cv2.HISTCMP\_INTERSECT: #교차 분석인 경우

            ret = ret/np.sum(query)        #비교대상으로 나누어 1로 정규화

        print("img%d:%7.2f"% (i+1 , ret), end='\t')

    print()

plt.show()

텍스트, 스크린샷, 모니터이(가) 표시된 사진

자동 생성된 설명