|  |  |
| --- | --- |
| 교육 제목 | Filtering |
| 교육 일시 | 2021. 11. 11 |
| 교육 장소 | 영우글로벌러닝 |
| **교육 내용** | |
| 오전 | import numpy as np  import sys  import cv2  import matplotlib.pyplot as plt  img1=cv2.imread('fig/lenna256.bmp', cv2.IMREAD\_GRAYSCALE)  img2=np.zeros((256, 256), np.uint8)\*255  cv2.circle(img2, (128, 128), 100, 120, -1)  cv2.circle(img2, (128, 128), 50, 50, -1)  dst1=cv2.add(img1, img2)  dst2=cv2.addWeighted(img1, 0.8, img2, 0.2, 0)  dst3=cv2.subtract(img1, img2)  dst4=cv2.absdiff(img1, img2)  # cv2.imshow('img1', img1)  # cv2.imshow('img2', img2)  # cv2.imshow('dst1', dst1)  # cv2.imshow('dst2', dst2)  # cv2.imshow('dst3', dst3)  # cv2.imshow('dst4', dst4)  # # cv2.waitKey()  # # cv2.destroyAllWindows()  plt.subplot(231), plt.axis('off'), plt.imshow(img1, 'gray'), plt.title('img1')  plt.subplot(232), plt.axis('off'), plt.imshow(img2, 'gray'), plt.title('img2')  plt.subplot(233), plt.axis('off'), plt.imshow(dst1, 'gray'), plt.title('dst1')  plt.subplot(234), plt.axis('off'), plt.imshow(dst2, 'gray'), plt.title('dst2')  plt.subplot(235), plt.axis('off'), plt.imshow(dst3, 'gray'), plt.title('dst3')  plt.subplot(236), plt.axis('off'), plt.imshow(dst4, 'gray'), plt.title('dst4')  plt.show()  src=cv2.imread('fig/flowers\_rgb.jpg', cv2.IMREAD\_COLOR)  src\_hsv=cv2.cvtColor(src, cv2.COLOR\_BGR2HSV)  h, s, v=cv2.split(src\_hsv)  cv2.imshow('src', src)  cv2.imshow('h', h)  cv2.imshow('s', s)  cv2.imshow('v', v)  cv2.waitKey()  cv2.destroyAllWindows()  src=cv2.imread('fig/lenna.bmp', cv2.IMREAD\_GRAYSCALE)  hist=cv2.calcHist([src], [0], None, [256], [0, 256])  plt.plot(hist, color='b')  plt.show()  cv2.imshow('src', src)  cv2.waitKey()  cv2.destroyAllWindows()  src=cv2.imread('fig/lenna.bmp')  src\_hsv=cv2.cvtColor(src, cv2.COLOR\_BGR2HSV)  h, s, v=cv2.split(src\_hsv)  hsv\_h = cv2.calcHist([h], [0], None, [256], [0, 256])  hsv\_s = cv2.calcHist([s], [0], None, [256], [0, 256])  hsv\_v = cv2.calcHist([v], [0], None, [256], [0, 256])  plt.plot(hsv\_h, color="b")  plt.plot(hsv\_s, color="g")  plt.plot(hsv\_v, color="r")  plt.show()  cv2.imshow('src', src)  cv2.waitKey()  cv2.destroyAllWindows()  src=cv2.imread('fig/field.bmp')  src\_YCrCb=cv2.cvtColor(src, cv2.COLOR\_BGR2YCrCb)  y, cr, cb=cv2.split(src\_YCrCb)  y=cv2.equalizeHist(y)  dst\_ycrcb=cv2.merge([y, cr, cb])  dst=cv2.cvtColor(dst\_ycrcb, cv2.COLOR\_YCrCb2BGR)  cv2.imshow('src', src)  cv2.imshow('dst', dst)  cv2.waitKey()  cv2.destroyAllWindows() |
| 오후 | import numpy as np  import sys  import cv2  src=cv2.imread('fig/Hawkes.jpg', cv2.IMREAD\_GRAYSCALE)  cv2.imshow('src', src)  dst\_equal=cv2.equalizeHist(src)  dst\_normal=cv2.normalize(src, None, 0, 255, cv2.NORM\_MINMAX, -1)  cv2.imshow('dst\_equal', dst\_equal)  cv2.imshow('dst\_normal', dst\_normal)  cv2.waitKey()  cv2.destroyAllWindows()  src=cv2.imread('fig/field.bmp', cv2.IMREAD\_COLOR)  src\_ycrcb=cv2.cvtColor(src, cv2.COLOR\_BGR2YCrCb)  y, cr, cb= cv2.split(src\_ycrcb)  y\_norm=cv2.normalize(y, None, 0, 255, cv2.NORM\_MINMAX, -1)  y\_equal=cv2.equalizeHist(y)  dst\_norm=cv2.merge([y\_norm, cr, cb])  dst\_equal=cv2.merge([y\_equal, cr, cb])  dst\_norm=cv2.cvtColor(dst\_norm, cv2.COLOR\_YCrCb2BGR)  dst\_equal=cv2.cvtColor(dst\_equal, cv2.COLOR\_YCrCb2BGR)  cv2.imshow('src', src)  cv2.imshow('dst\_norm', dst\_norm)  cv2.imshow('dst\_equal', dst\_equal)  cv2.waitKey()  cv2.destroyAllWindows()  src=cv2.imread('fig/candies2.png')  src\_hsv=cv2.cvtColor(src, cv2.COLOR\_BGR2HSV)    cv2.imshow('src', src)  # inRange(src, lowerb, upperb[, dst]) -> dst  # src: 입력영상  # lowerb: 하한값  # upperb: 상한값  dst1=cv2.inRange(src, (0, 128, 0), (100, 255, 100))  dst2=cv2.inRange(src\_hsv, (50, 170, 0), (80, 255, 255))  cv2.imshow('dst1', dst1)  cv2.imshow('dst2', dst2)  cv2.waitKey()  cv2.destroyAllWindows()  src=cv2.imread('fig/flowers.jpg')  src\_hsv=cv2.cvtColor(src, cv2.COLOR\_BGR2HSV)    cv2.imshow('src', src)  dst1=cv2.inRange(src, (0, 128, 0), (100, 255, 100))  dst2=cv2.inRange(src\_hsv, (50, 170, 0), (80, 255, 255))  cv2.imshow('dst1', dst1)  cv2.imshow('dst2', dst2)  cv2.waitKey()  cv2.destroyAllWindows()  src=cv2.imread('fig/cropland.png')  x, y, w, h=cv2.selectROI(src)  src\_ycrcb=cv2.cvtColor(src, cv2.COLOR\_BGR2YCrCb)  crop=src\_ycrcb[y:y+h, x:x+w]  # calcHist(images, channels, mask, histSize, ranges[ , hist[ , accumulate]]) -> hist  hist=cv2.calcHist([crop], [1,2], None, [128, 128], [0, 256, 0, 256])  # calcBackProject(images, channels, hist, ranges, scale[, dst]) -> dst  backproj=cv2.calcBackProject([src\_ycrcb], [1,2], hist, [0, 256, 0, 256], 1)  dst=cv2.copyTo(src, backproj)  cv2.imshow('src', src)  cv2.imshow('backproj', backproj)  cv2.imshow('dst', dst)  cv2.waitKey()  cv2.destroyAllWindows()  src=cv2.imread('fig/blue\_eyes.png', cv2.IMREAD\_GRAYSCALE)  # cv2.filter2D(src, ddepth, kernel[, dst[, anchor[, delta[, borderType]]]]) -> dst  # src: 입력영상  # ddepth: 출력영상의 타입 (cv2.CV\_8U, cv2.CV\_32F, cv2.CV\_64F),  # -1 => 입력영상과 같은 타입  # kernel: filter 행렬, 실수형  # anchor: (-1, -1) 필터의 중앙점  # delta: 더할 값  # borderType: 가장자리 픽셀확장 방식  # dst: 출력영상  kernel\_3=np.ones((3,3), np.float32)/9  kernel\_5=np.ones((5,5), np.float32)/25  src\_mean\_filter3=cv2.filter2D(src, -1, kernel\_3) #, (-1, -1), 0)  src\_mean\_filter5=cv2.filter2D(src, -1, kernel\_5)  cv2.imshow('src', src)  cv2.imshow('src\_mean\_filter3', src\_mean\_filter3)  cv2.imshow('src\_mean\_filter5', src\_mean\_filter5)  cv2.waitKey()  cv2.destroyAllWindows() |