|  |  |
| --- | --- |
| 교육제목 | 데이터 기반 인공지능 시스템 엔지니어 양성 과정 |
| 교육일시 | 211111 |
| 교육장소 | YGL 학과장 및 자택(디스코드 사용 온라인 학습) |
| **교육내용** | |
| 1. 영상에서 원하는 색깔만 지정해서 뽑아내기   # 캔디에서 원하는 색깔만 뽑아내보자  src = cv2.imread('fig/candies2.png')  # inRange(src, lowerb, upperb[, dst]) -> dst  # src ; 입력영상  # lowerb : 하한값  # upperb : 상한값  src\_hsv = cv2.cvtColor(src, cv2.COLOR\_BGR2HSV)  dst1 = cv2.inRange(src,(0, 128, 0),(100, 255, 100)) # b,g,r 채널  dst2 = cv2.inRange(src\_hsv,(50, 170, 0),(80, 255, 255)) # b,g,r 채널  cv2.imshow('src',src)  cv2.imshow('dst1',dst1)  cv2.imshow('dst2',dst2)  cv2.waitKey()  cv2.destroyAllWindows()   1. 영상에서 원하는 Area 직접 지정해서 뽑기   src = cv2.imread('fig/flowers.jpg')  x, y, w, h = cv2.selectROI(src)  src\_ycrcb = cv2.cvtColor(src, cv2.COLOR\_BGR2YCrCb) # numpy array 타입  # 시작점부터 박스 드랍하는곳...지정  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, [64, 64], [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()   1. Filter   import numpy as np  import sys  import cv2  # 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  # blur(src, ksize[, dst[, anchor[, borderType]]]) -> dst  # src: 입력영상  # ksize: mean filter kernel size  import sys  import numpy as np  import cv2  src = cv2.imread('fig/blue\_eyes.png', cv2.IMREAD\_GRAYSCALE)  if src is None:  print('Image load failed!')  sys.exit()  #kernel\_3 = np.ones((3, 3), dtype=np.float64) / 9.  # kernel\_5 = np.ones((5, 5), dtype=np.float64) / 25.  # dst = cv2.filter2D(src, -1, kernel\_5)  # dst = cv2.blur(src, (3, 3))  cv2.imshow('src', src)  for ksize in (3, 5, 7, 9):  dst = cv2.blur(src, (ksize, ksize))    text = '{} x {}'. format(ksize, ksize)  cv2.putText(dst, text, (10, 30), cv2.FONT\_HERSHEY\_SIMPLEX,  1, 0, 2, cv2.LINE\_AA)    cv2.imshow('dst', dst)  cv2.waitKey()  # cv2.imshow('dst', dst)  # cv2.waitKey()  cv2.destroyAllWindows() | |