|  |  |
| --- | --- |
| 교육제목 | 데이터 기반 인공지능 시스템 엔지니어 양성 과정 |
| 교육일시 | 211109 |
| 교육장소 | YGL 학과장 및 자택(디스코드 사용 온라인 학습) |
| **교육내용** | |
| 1. Image segmentation(영상 분할)   목적 : 전체 영상을 객체 또는 의미 있는 객체의 일부 영역 집합으로 구분  필요성 : 영상에 포함된 객체나 관심 영역에 대한 구분  픽셀 단위가 아닌 고수준에서의 영상 처리 수행 가능   * Segmaentation의 기본 방법들   영역 내부의 유사성 측정  영역간의 차이 측정( 밝기, 색, 질감 등의 특징을 사용)   * 고려 요소   노이즈  연결성( 공간적으로 디지털화한 결과에서 기인함)  캡처.JPG   1. Morphology(형태학)   생물학의 한 분야로 동물이나 식물의 모양이나 구조를 다루는 학문  관심 객체의 검출을 쉽게 처리할 수 있도록 영상 분할 결과를 단순화하는 방법으로 사용.  객체 경계의 단순화, 작은 구멍을 채움, 작은 돌기의 제거 등  Binary 영상과 Gray-scale영상에 적용 가능  모폴로지 필터링 : 구조적 요소와 팽창 및 침식 연산 사용   1. 팽창 연산   객체의 크기를 확장 : 객체 내부의 작은 구멍을 채움, 근접한 위치의 두 객체를 연결  캡처.JPG  캡처.JPG  캡처.JPG   1. 침식연산   객체의 크기를 축소 : 객체 경계를 침식, 작은 돌기를 제거  캡처.JPG  캡처.JPG  캡처.JPG   1. 키를 눌러 색 반전시키기   import numpy as np  import sys  import cv2  # 키를 눌러 색 반전시키기  img = cv2.imread('fig/candies.png')  if img is None:  print('Image load failed!')  sys.exit()    cv2.imshow('img', img)  while True:    key = cv2.waitKey()  if key == 27 or key == ord('q'):  break    elif key == ord('i'): # i를 누르면 색이 뒤집힌다  img = cv2.bitwise\_not(img)  cv2.imshow('img', img)    cv2.destroyAllWindows()  # 그레이스케일로 반전시키기  img = cv2.imread('fig/candies.png' , cv2.IMREAD\_GRAYSCALE)  if img is None:  print('Image load failed!')  sys.exit()    cv2.imshow('img', img)  while True:    key = cv2.waitKey()  if key == 27 or key == ord('q'):  break    elif key == ord('i'): # i를 누르면 색이 뒤집힌다  # img = cv2.bitwise\_not(img)  img = ~img  cv2.imshow('img', img)    cv2.destroyAllWindows()   1. 마우스에서 입력 받기   # 마우스 함수 선언  oldx = -1  oldy = -1  def call\_mouse(event, x, y, flags, param):  global oldx, oldy    if event == cv2.EVENT\_LBUTTONDOWN: # 왼쪽 버튼을 누를 때  # print('EVENT\_LBUTTONDOWN = {},{}'.format(x, y))  oldx, oldy = x, y  # elif event == cv2.EVENT\_LBUTTONUP: # 왼쪽 버튼을 뗄 때  # print('EVENT\_LBUTTONUP = {},{}'.format(x, y))      elif event == cv2.EVENT\_MOUSEMOVE:  if flags == cv2.EVENT\_FLAG\_LBUTTON:  cv2.line(img, (oldx, oldy),(x,y), (0, 0, 255), 4, cv2.LINE\_AA)  cv2.imshow('img', img)  oldx, oldy = x, y    img = np.ones((400,600,3), dtype=np.uint8)\*255 # 검은색 캔버스를 흰색으로 바꾼다  cv2.namedWindow('img')  # 마우스콜백  # cv2.setMouseCallback(windowName, onMouse, param = None) -> None  # windowName: 마우스이벤트를 수행할 창 이름  # onMouse: 마우스 이벤트 콜벡함수  # param: 콜백함수에 전달할 데이터  cv2.setMouseCallback('img', call\_mouse, img)  cv2.imshow('img',img)  cv2.waitKey()  cv2.destroyAllWindows()   1. 트랙바 – 알파채널 그림파일   # 알파채널 활용해서 해보자  def call\_track(pos):  img\_glass = img\*pos # 제일 처음값은 1..이니까 0이랑곱하는건 쭉 0  cv2.imshow('img', img\_glass)  img\_alpha = cv2.imread('fig/imgbin\_sunglasses\_1.png', cv2.IMREAD\_UNCHANGED)  img = img\_alpha[:,:,-1]  img[img>0] = 1 # 0이상인값은 전부 1로  cv2.namedWindow('img')  cv2.createTrackbar('level', 'img', 0, 255, call\_track)  cv2.imshow('img', img)  cv2.waitKey()  cv2.destroyAllWindows()   1. 영상에 일정 숫자를 더해서 밝기 올리기   import numpy as np  import sys  import cv2  src = cv2.imread('fig/lenna.bmp', cv2.IMREAD\_COLOR)  if src is None:  print('image read faild')  sys.exit()    # dst = src + 100 # 밝은곳에 100을 더하니까 255를 넘김  dst = cv2.add(src, (100, 100, 100, 0))  # dst = np.clip(src + 100., 0, 255).astype(np.uint8) # 255보다 커지면 255로 고정  cv2.imshow('src',src)  cv2.imshow('dst',dst)  cv2.waitKey()  cv2.destroyAllWindows() | |