|  |  |
| --- | --- |
| 교육 제목 | Image |
| 교육 일시 | 2021. 11. 8 |
| 교육 장소 | 영우글로벌러닝 |
| **교육 내용** | |
| 오전 | import numpy as np  import sys  import cv2  img1=cv2.imread('fig/puppy.bmp', cv2.IMREAD\_COLOR)  img2=img1  img3=img1.copy()  img1[200:300, 200:300]  cv2.imshow('img1', img1)  cv2.imshow('img2', img2)  cv2.imshow('img3', img3)  cv2.waitKey()  cv2.destroyAllWindows()  src=cv2.imread('fig/airplane.bmp')  mask=cv2.imread('fig/mask\_plane.bmp', cv2.IMREAD\_GRAYSCALE)  dst=cv2.imread('fig/field.bmp')  cv2.copyTo(src, mask, dst)  cv2.imshow('src', src)  cv2.imshow('mask', mask)  cv2.imshow('dst', dst)  cv2.waitKey()  cv2.destroyAllWindows()  src=cv2.imread('fig/puppy.bmp', cv2.IMREAD\_COLOR)  sunglass=cv2.imread('fig/imgbin\_sunglasses\_1.png', cv2.IMREAD\_UNCHANGED)  sunglass=cv2.resize(sunglass, (300, 150))  mask=sunglass[:, :, -1]  glass=sunglass[:, :, 0:3]  h, w= mask.shape  crop=src[120:120+h, 220:220+w]  # cv2.copyTo(glass, mask, crop)  crop[mask>0]=(0, 0, 255)  cv2.imshow('src', src)  cv2.imshow('mask', mask)  cv2.imshow('crop', crop)  cv2.waitKey()  cv2.destroyAllWindows() |
| 오후 | img=np.ones((600, 1200, 3), dtype=np.uint8)\*255  # cv2.line(img, pt1, pt2, color, thickness=None),  # lineType=None, shift=None) -> img  # flage  # img : 그림을 그릴 영상  # pt1, pt2 : 직선의 시작점, 끝점  # color : 직선의 칼라 (B,G,R)의 튜플  # thickness : 선두께, 기본=1  # lineType : cv2.LINE\_4, cv2.LINE\_8, cv2.LINE\_AA  cv2.line(img, (50,50), (300, 50), (0, 0, 255), 1, cv2.LINE\_8)  # cv2.rectangle(img, pt1, pt2, color, thickness=None, lineType=None) -> img  # pt1 : 좌측상단, pt2 : 우측하단  # cv2.rectangle(img, rect, color, thickness=None, lineType=None) -> img  # rect : 사각형의 위치 정보 (x, y, w, h)    cv2.rectangle(img, (50, 400), (400, 500), (0, 255, 0), -1)  # cv2.circle(img, center, radius, color, thickness=None, lineType=none) -> img  # center : 원의 중심좌표 (x, y)  # radius : 원의 반지름  cv2.circle(img, (600, 200), 100, (0, 0, 255), -1)  cv2.imshow('canvas', img)  cv2.waitKey()  cv2.destroyAllWindows()  img=np.ones((600, 1200, 3), dtype=np.uint8)\*255  # 인  cv2.circle(img, (150, 150), 100, (0, 0, 255), 2, cv2.LINE\_AA)  cv2.line(img, (300,50), (300, 250), (0, 0, 255), 2, cv2.LINE\_8)  cv2.line(img, (100,300), (100, 400), (0, 0, 255), 2, cv2.LINE\_8)  cv2.line(img, (100,400), (300, 400), (0, 0, 255), 2, cv2.LINE\_8)  # 삼  cv2.line(img, (500,50), (400, 250), (0, 0, 255), 2, cv2.LINE\_AA)  cv2.line(img, (500,50), (600, 250), (0, 0, 255), 2, cv2.LINE\_AA)  cv2.line(img, (650,50), (650, 250), (0, 0, 255), 2, cv2.LINE\_8)  cv2.line(img, (650,125), (700, 125), (0, 0, 255), 2, cv2.LINE\_8)  cv2.rectangle(img, (450, 300), (600, 400), (0, 0, 255), 2)  cv2.imshow('canvas', img)  cv2.waitKey()  cv2.destroyAllWindows()  cap=cv2.VideoCapture(0)  if not cap.isOpened():  print('Video open faild')  cap.release()  sys.exit()  # a='DIVX'  # print(a)  # print(\*a)  width=int(cap.get(cv2.CAP\_PROP\_FRAME\_WIDTH))  height=int(cap.get(cv2.CAP\_PROP\_FRAME\_HEIGHT))  fps=cap.get(cv2.CAP\_PROP\_FPS)  fourcc=cv2.VideoWriter\_fourcc('D', 'I', 'V', 'X')  # fourcc=cv2.VideoWriter\_fourcc(\*DIVX)  # print(type(width))  # print('width and height = {} x {}'.format(width, height))  # print(fps)  # cv2.VideoWriter(filename, fourcc, fps, framesize, isColor=None)  out=cv2.VideoWriter('webcam\_record\_avi', fourcc, fps, (width, height))  while True:  # cv2.VideoCapture.read(imge=None) -> retval, image  # retval : True or False  # image : Frame  ret, frame=cap.read()    if not ret:  print('Video read failed')  break    # frame=cv2.Canny(frame, 50, 150)    cv2.imshow('cap', frame)  out.write(frame)    if cv2.waitKey(10) ==27:  break    cap.release()  out.release()  cv2.destroyAllWindows() |