|  |  |
| --- | --- |
| 교육 제목 |  |
| 교육 일시 | 2021. 11. 10 |
| 교육 장소 | 영우글로벌러닝 |
| **교육 내용** | |
| 오전 | import numpy as np  import sys  import cv2  cap=cv2.VideoCapture(0)  if not cap.isOpened():  print('Video open failed')  cap.release()  sys.exit()    cv2.namedWindow('Webcam')  width=int(cap.get(cv2.CAP\_PROP\_FRAME\_WIDTH))  height=int(cap.get(cv2.CAP\_PROP\_FRAME\_HEIGHT))  fps=int(cap.get(cv2.CAP\_PROP\_FPS))  fourcc=cv2.VideoWriter\_fourcc(\*'DIVX')  out=cv2.VideoWriter('my.avi', fourcc, fps, (width, height))  while True:  ret, frame=cap.read()    if not ret:  print('Video read failed')  break    edge=cv2.Canny(frame, 50, 100)  edge=cv2.cvtColor(edge, cv2.COLOR\_GRAY2BGR)  out.write(edge)    cv2.imshow('Webcam', frame)  cv2.imshow('edge', edge)    if cv2.waitKey(10)==27:  break    cap.release()  out.release()  cv2.destroyAllWindows()  def call\_mouse(event, x, y, flags, param):  global oldx, oldy  if event==cv2.EVENT\_LBUTTONDOWN:  # print('EVENT\_LBUTTONDOWN : ', x, y)  oldx, oldy=x,y  # elif event==cv2.EVENT\_LBUTTONUP:  # print('EVENT\_LBUTTONUP : ', x, y)  elif event==cv2.EVENT\_MOUSEMOVE:  if flags==cv2.EVENT\_FLAG\_LBUTTON:  cv2.line(img, (oldx, oldy), (x, y), (0, 0, 255), 6, cv2.LINE\_AA)  cv2.imshow('img', img)  oldx, oldy=x, y      img=np.ones((400, 600, 3), dtype=np.uint8)\*255  cv2.namedWindow('img')  cv2.setMouseCallback('img', call\_mouse, img)  cv2.imshow('img', img)  cv2.waitKey()  cv2.destroyAllWindows()  def call\_track(pos):  img[:]=pos  cv2.imshow('img', img)    img=np.zeros((400, 600, 3), dtype=np.uint8)  cv2.namedWindow('img')  cv2.createTrackbar('level', 'img', 5, 255, call\_track)  cv2.imshow('img', img)  cv2.waitKey()  cv2.destroyAllWindows()  src=cv2.imread('fig/lenna.bmp', cv2.IMREAD\_COLOR)  if img is None:  print('Image load failed!')  sys.exit()  # dst=src+100  dst=cv2.add(src, (0, 0, 100, 0))  # dst=np.clip(src+100., 0, 255).astype(np.uint8)  # dst=cv2.cvtColor(src, cv2.COLOR\_BGR2HSV) # 명암만 올리기  cv2.imshow('img', src)  cv2.imshow('dst', dst)    cv2.waitKey()  cv2.destroyAllWindows()  img1=cv2.imread('fig/lenna256.bmp', cv2.IMREAD\_GRAYSCALE)  img2=np.zeros((256, 256), np.uint8)  cv2.circle(img2, (128, 128), 100, 100, -1)  cv2.circle(img2, (128, 128), 50, 50, -1)  dst1=cv2.add(img1, img2)  dst2=cv2.addWeighted(img1, 0.2, img2, 0.8, 0.0)  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() |
| 오후 | src=cv2.imread('fig/flowers.jpg', cv2.IMREAD\_COLOR)  print(src.shape)  print(type(src))  # b, g, r=cv2.split(src)  # b=src[:, :, 0]  # g=src[:, :, 1]  # r=src[:, :, 2]  src\_hsv=cv2.cvtColor(src, cv2.COLOR\_BGR2HSV)  h, s, v=cv2.split(src\_hsv)  # h=np.clip(h\*2., 0, 255).astype(np.uint8)  cv2.imshow('src', src)  # cv2.imshow('b\_channel', b)  # cv2.imshow('g\_channel', g)  # cv2.imshow('r\_channel', r)  cv2.imshow('h\_channel', h)  cv2.imshow('s\_channel', s)  cv2.imshow('v\_channel', v)  cv2.waitKey()  cv2.destroyAllWindows()  import matplotlib.pyplot as plt  src=cv2.imread('fig/lenna.bmp')  # cv2.calcHist(images, channels, mask, histSize, ranges[, hist[, accumulate]]) -> hist  # images: 입력영상 리스트(리스트로 입력)  # channels: 채널리스트, 3 채널의 경우 [0,1,2]  # mask: 마스크 영상입력, 영상전체는 None으로 지정  # histSize: 히스터그램 빈의 크기  # range: 히스토그램의 최솟값과 최댓값  # hist: 계산된 히스토그램, numpy.ndarray 타입  # accumulate: 기존의 히스토그램을 누적할경우 True  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])  cv2.imshow('src', src)  cv2.waitKey()  cv2.destroyAllWindows()  plt.plot(hsv\_h, color="b")  plt.plot(hsv\_s, color="g")  plt.plot(hsv\_v, color="r")  plt.show()  ------------------------------------------------------------------------------------  import numpy as np  import sys  import cv2  src=cv2.imread('fig/Hawkes.jpg', cv2.IMREAD\_GRAYSCALE)  cv2.imshow('src', src)  # smin, smax, \_, \_=cv2.minMaxLoc(src)  # dst=np.clip(255\*(src-smin)/(smax-smin), 0, 255).astype(np.uint8)  # print('min', smin)  # print('max', smax)  # cv2.normalize(src, dst=None, alpha=None, beta=None, norm\_type=None, dtype=None, mask=None) -> dst  # src: 입력영상  # dst: 결과영상  # alpha: 정규화 최소값 (예, 0)  # beta: 정규화 최댓값 (예, 155)  # norm\_type: cv2.NORM\_MINMAX  # dtype = -1, src와 같은 type  dst=cv2.normalize(src, None, 0, 255, cv2.NORM\_MINMAX, -1)  cv2.imshow('dst', dst)  cv2.waitKey()  cv2.destroyAllWindows()  src=cv2.imread('fig/Hawkes.jpg', cv2.IMREAD\_GRAYSCALE)  cv2.imshow('src', src)  dst\_equal=cv2.equalizeHist(src)  dst\_nomal=cv2.normalize(src, None, 0, 255, cv2.NORM\_MINMAX, -1)  cv2.imshow('dst\_equal', dst\_equal)  cv2.imshow('dst\_nomal', dst\_nomal)  cv2.waitKey()  cv2.destroyAllWindows()  src=cv2.imread('fig/field.bmp')  # src\_hsv=cv2.cvtColor(src, cv2.COLOR\_BGR2HSV)  src\_YCrCb=cv2.cvtColor(src, cv2.COLOR\_BGR2YCrCb)  # src\_split=cv2.split(src\_YCrCb)  # src\_split[0]=cv2.equalizeHist(src\_split[0])  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() |