|  |  |
| --- | --- |
| 실습 8 (고양이 이미지) | 소스코드 및 결과이미지  import cv2  import numpy as np  image = cv2.imread('C:/Users/khy18/Downloads/img/cat.jpg')  image = cv2.blur(image, (7,7))  gray = cv2.cvtColor(image, cv2.COLOR\_BGR2GRAY)  \_, binary = cv2.threshold(gray, 110, 255, cv2.THRESH\_BINARY)  kernel = np.ones((3,3), np.uint8)  eroded\_image = cv2.erode(binary, kernel, iterations = 6)  kernel\_erode = np.ones((3, 3), np.uint8)  dilated\_image = cv2.dilate(eroded\_image,kernel\_erode, iterations =7)  cv2.imshow('img/cat\_seg.png', dilated\_image)  cv2.waitKey(0)  cv2.destroyAllWindows() |
| 실습 9 (팝아트 이미지) | 소스코드 및 결과이미지  import cv2  import numpy as np  image = cv2.imread('C:/Users/khy18/Downloads/img/face.jpg')  image = cv2.blur(image, (3,3))  gray = cv2.cvtColor(image, cv2.COLOR\_BGR2GRAY)  \_, binary = cv2.threshold(gray, 160, 255, cv2.THRESH\_BINARY)  cv2.imshow('Processed binary Image', binary)  red = np.full((image.shape[0],image.shape[1],3), 255, dtype='uint8')  red[binary==0] = (0,0,255)  cv2.imshow('red',red)  blue = np.full((image.shape[0],image.shape[1],3), 255, dtype='uint8')  blue[binary==0] = (255,0,0)  cv2.imshow('blue',blue)  green = np.full((image.shape[0],image.shape[1],3), 255, dtype='uint8')  green[binary==0] = (0,255,0)  cv2.imshow('green',green)  black = np.full((image.shape[0],image.shape[1],3), 255, dtype='uint8')  black[binary==0] = (0,0,0)  c1 = cv2.hconcat([black, red])  c2 = cv2.hconcat([blue, green])  c3 = cv2.vconcat([c1,c2])  cv2.waitKey(0)  cv2.destroyAllWindows() |
| 실습 10 (블록 바운더리) | 소스코드 및 결과이미지  import cv2  import numpy as np  img = cv2.imread('C:/Users/khy18/Downloads/img/block.jpg')  # HSV 색공간으로 변환 및 분리  hsv = cv2.cvtColor(img, cv2.COLOR\_BGR2HSV)  h, s, v = cv2.split(hsv)  # s(채도)와 v(밝기)공간에 대해 adaptive threshold 적용하여 이진화  thresh0 = cv2.adaptiveThreshold(s, 255,  cv2.ADAPTIVE\_THRESH\_MEAN\_C, cv2.THRESH\_BINARY\_INV, 19, 11)  thresh1 = cv2.adaptiveThreshold(v, 255,  cv2.ADAPTIVE\_THRESH\_MEAN\_C, cv2.THRESH\_BINARY\_INV, 21, 13)  # binary 이미지 통합  thresh = cv2.bitwise\_or(thresh0, thresh1)  cv2.imshow('Image-thresh', thresh)  cv2.imshow('Image-thresh0', thresh0)  cv2.imshow('Image-thresh1', thresh1)  kernel = np.ones((3, 3), np.uint8)  img = cv2.dilate(thresh, kernel, iterations=1)  cv2.imshow('1', img)  kernel = np.ones((3, 3), np.uint8)  img = cv2.erode(img, kernel, iterations=2)  cv2.imshow('2', img)  kernel = np.ones((5, 5), np.uint8)  img = cv2.dilate(img, kernel, iterations=3)  cv2.imshow('3', img)  img = cv2.erode(img, kernel, iterations=3)  cv2.imshow('4', img)  cv2.imwrite('img/box\_seg.png',img)  cv2.waitKey(0)  cv2.destroyAllWindows() |
| 실습 11 (Edge) | 소스코드 및 결과이미지  import cv2  import numpy as np  # 이미지 불러오기  img = cv2.imread('C:/Users/khy18/Downloads/img/diary.png')  img\_contour = img.copy()  # 그레이스케일로 변환  gray = cv2.cvtColor(img, cv2.COLOR\_BGR2GRAY)  # 이진화  \_, thresh = cv2.threshold(gray, 150, 255, cv2.THRESH\_BINARY)  thresh = cv2.bitwise\_not(thresh)  binary\_img\_line = np.zeros\_like(thresh)  # 캐니 에지 검출  edges = cv2.Canny(gray, 0, 150, apertureSize=3)  cv2.imshow('edges', edges)  # 허프 라인 변환  lines = cv2.HoughLines(edges, 1, np.pi/180, 230)  for line in lines:      rho, theta = line[0]  a = np.cos(theta)  b = np.sin(theta)  x0 = a \* rho  y0 = b \* rho  x1 = int(x0 + 1000\*(-b))  y1 = int(y0 + 1000\*(a))  x2 = int(x0 - 1000\*(-b))  y2 = int(y0 - 1000\*(a))  cv2.line(binary\_img\_line, (x1, y1), (x2, y2), (255), 1)  cv2.imshow('hough line', binary\_img\_line)  # 컨투어 찾기  contours, \_ = cv2.findContours(thresh, cv2.RETR\_LIST, cv2.CHAIN\_APPROX\_SIMPLE)  binary\_img = np.zeros\_like(thresh)  cv2.drawContours(binary\_img, contours, -1, (255), 2) # 마지막 인자는 선 두께cv2.drawContours(img\_contour, contours, -1, (0, 255, 0), 3)  cv2.imshow('Contours', binary\_img)  empty\_note\_bw = cv2.bitwise\_and(binary\_img, binary\_img\_line)  cv2.imshow('empty\_note\_bw', empty\_note\_bw)  empty\_note\_color = img.copy()  empty\_note\_color[empty\_note\_bw==0] = (255,255,255)  cv2.imshow('empty\_note\_color',empty\_note\_color)  cv2.waitKey(0)  cv2.destroyAllWindows() |
| 실습 12 (Grabcut) | 소스코드 및 결과이미지  import cv2  import numpy as np  drawing = False  ix, iy = -1, -1  rect\_over = False  x, y = -1, -1  # 마우스 콜백 함수  def draw\_rectangle(event, \_x, \_y, flags, param):      global ix, iy, drawing, img, img2, rect\_over, x, y # x, y를 전역 변수로 추가      if event == cv2.EVENT\_LBUTTONDOWN:          drawing = True          ix, iy = \_x, \_y # \_x, \_y 사용      elif event == cv2.EVENT\_LBUTTONUP:          drawing = False          cv2.rectangle(img, (ix, iy), (\_x, \_y), (0, 255, 0), 2)          rect\_over = True          x, y = \_x, \_y # 마우스 뗄 때 위치 전역 변수저장  img = cv2.imread('C:/Users/khy18/Downloads/img/box.jpg')  img2 = img.copy() # 이미지 복사  cv2.namedWindow('image')  cv2.setMouseCallback('image', draw\_rectangle)  while(1):      cv2.imshow('image', img)      k = cv2.waitKey(1) & 0xFF      if k == 27: # ESC 키 누르면 종료          break      elif k == ord('r') and rect\_over: # r 키 누르면 GrabCut 적용          mask = np.zeros(img.shape[:2],np.uint8)          bgd\_model = np.zeros((1,65),np.float64)          fgd\_model = np.zeros((1,65),np.float64)          rect = (min(ix,x),min(iy,y),abs(ix-x),abs(iy-y))          cv2.grabCut(img2,mask,rect,bgd\_model,fgd\_model,5,cv2.GC\_INIT\_WITH\_RECT)          mask2 = np.where((mask==2)|(mask==0),0,1).astype('uint8')          img = img2\*mask2[:,:,np.newaxis]          rect\_over = False  cv2.destroyAllWindows() |