# #1. Day1

## ##1. opencv.py

import cv2

# 실행: 컨트롤 + 쉬프트 + F10

print('현재 OpenCV 버전: ', cv2.\_\_version\_\_)

# 그레이스케일 영상 가져오기

#img = cv2.imread('./dog.bmp', cv2.IMREAD\_GRAYSCALE)

#print(img)

# 트루컬러 영상

img = cv2.imread('./dog.bmp')

print(img)

cv2.imshow('img', img) # 창이름, 영상

cv2.waitKey() # 키를 입력할 때까지 계속 대기

## ##2. matplotlib.py

import cv2

import matplotlib.pyplot as plt

'''

# cv2를 통해 출력

img = cv2.imread('./dog.bmp', cv2.IMREAD\_GRAYSCALE)

cv2.imshow('img', img)

cv2.waitKey()

# matplotlib을 통해 그레이스케일 출력

img = cv2.imread('./dog.bmp', cv2.IMREAD\_GRAYSCALE)

plt.axis('off')

plt.imshow(img, cmap='gray')

plt.show()

img = cv2.imread('./dog.bmp')

plt.axis('off')

plt.imshow(img)

plt.show()

# matplotlib을 통해 트루컬러 출력(cvtColor 사용)

img = cv2.imread('./dog.bmp')

img = cv2.cvtColor(img, cv2.COLOR\_BGR2RGB)

plt.axis('off')

plt.imshow(img)

plt.show()

'''

img\_gray = cv2.imread('./dog.bmp', cv2.IMREAD\_GRAYSCALE)

img\_color = cv2.imread('./dog.bmp')

img\_color = cv2.cvtColor(img\_color, cv2.COLOR\_BGR2RGB)

plt.subplot(121)

plt.axis('off')

plt.imshow(img\_gray, cmap='gray')

plt.subplot(122)

plt.axis('off')

plt.imshow(img\_color)

plt.show()

## ## 3. imginfo.py

import cv2

'''

img\_gray = cv2.imread('./dog.bmp',cv2.IMREAD\_GRAYSCALE)

print('img\_gray type:', type(img\_gray))

print('img\_gray shape:', img\_gray.shape) # h / w 픽셀

print('img\_gray dtype:', img\_gray.dtype)

img\_color = cv2.imread('./dog.bmp')

print('img\_gray type:', type(img\_color))

print('img\_gray shape:', img\_color.shape) # h / w 픽셀 채널 3개

print('img\_gray dtype:', img\_color.dtype)

# img\_color의 정보를 아래와 같이 출력

# img\_color 사이즈 : 548\*364

img\_color = cv2.imread('./dog.bmp')

# 개인 답안

he, wi, ch = list(img\_color.shape)

print(f'img\_color 사이즈: {wi}\*{he}')

# 강사님 답안

h,w = img\_color.shape[:2]

print(f'img\_color 사이즈: {w}\*{h}')

'''

# 문제2

# img\_gray가 그레이스케일 영상인지 컬러 영상인지 구별하는 프로그램을 작성

# if문 사용

# img\_gray는 그레이스케일 영상입니다

img\_color = cv2.imread('./dog.bmp')

img\_gray = cv2.imread('./dog.bmp',cv2.IMREAD\_GRAYSCALE)

img = img\_color

# 개인답안

if len(img.shape) == 3:

print('img는 컬러 영상입니다')

else:

print('img는 그레이스케일 영상입니다')

# 강사님답안

if len(img\_gray.shape) == 2:

print('img\_gray 그레이스케일 영상입니다')

elif len(img\_gray.shape) == 3:

print('img\_gray 컬러 영상입니다')

# 문제3

# img\_color 특정 색 정보로 영상을 출력

# BGR: (255,102, 255)

img\_color = cv2.imread('./dog.bmp')

# 개인답변 - 더 좋은 답안

img\_color[:,:] = (255,102, 255)

'''

# 강사님

for x in range(h):

for y in range(w):

img\_color[x,y] = (255,102,255)

'''

cv2.imshow('img', img\_color) # 창이름, 영상

cv2.waitKey() # 키를 입력할 때까지 계속 대기

## ## 4. createimage.py

import cv2

import numpy as np

img1 = np.empty((240,320), dtype=np.uint8) # 픽셀 사이즈 / 그레이스케일

img2 = np.zeros((240,320,3), dtype=np.uint8)

img3 = np.ones((240,320), dtype=np.uint8) \* 120 # 밝기정보 120

img4 = np.full((240,320,3), (255,102,255), dtype=np.uint8) # 색상정보

cv2.imshow('img1', img1)

cv2.imshow('img2', img2)

cv2.imshow('img3', img3)

cv2.imshow('img4', img4)

cv2.waitKey()

## ## 5. copyimage.py

import cv2

img = cv2.imread('./dog.bmp')

img\_copy = img.copy() # 복사하면 서로 다른 공간을 할당함 -> 결과가 다르다!

# img\_test = img # 할당하게 되면 같은 공간을 가리킴 -> 결과가 같다!

img\_copy[91:210, 125:245] = (244,102,255)

cv2.imshow('img', img)

cv2.imshow('img\_copy', img\_copy)

cv2.waitKey()

# #2. Day2

## ##1. drawing.py

import cv2

import numpy as np

# 높이500 / 너비500 / 3개의 채널가지고 값을 255로 채움

img = np.full((500,500,3), 255, np.uint8)

# 시작점(70,70)튜플 -> 끝점(250,70), 색상:빨강(BGR) ,두께5

cv2.line(img, (70,70), (250,70), (0,0,255), 5)

cv2.rectangle(img, (50,200,150,100),(0,255,0),3)

cv2.circle(img, (300,100), 50,(255,255,0),-1)

str = 'Hello OpenCV!'

# 넣을 이미지, 넣을 문자, 넣을 좌표, 글꼴, 폰트크기, 색상

cv2.putText(img, str, (30,300), cv2.FONT\_HERSHEY\_SIMPLEX,2,(0,0,255))

cv2.imshow('img',img)

cv2.waitKey()

## ##2. camerain.py

import cv2

import sys

# 카메라를 여는 방법\_기본카메라:0 / 별도의 카메라는 고유의 INDEX값(VideoCapture객체생성)

cap = cv2.VideoCapture(0)

# 카메라에 연결되었는지(True:성공)

if not cap.isOpened():

print('카메라를 열 수 없습니다')

sys.exit() # 카메라 종료

print('카메라 연결 성공!')

print('가로사이즈', int(cap.get(cv2.CAP\_PROP\_FRAME\_WIDTH))) # 가로

print('세로사이즈', int(cap.get(cv2.CAP\_PROP\_FRAME\_HEIGHT))) # 세로

while True:

# ret:리턴되었는지여부(True) frame:카메라의 프레임

ret, frame = cap.read()

# 리턴이 안되었으면 종료

if not ret:

break

# frame을 출력하기

cv2.imshow('frame', frame)

# 10ms마다 입력을 확인하면 esc(27)이 눌리면 종료

if cv2.waitKey(10) == 27: # ESC

break

# VideoCapture객체 해제 : 카메라 연결해제

cap.release()

## ##3. videoin.py

import cv2

import sys

cap = cv2.VideoCapture('./tiger.mp4')

if not cap.isOpened():

print('동영상을 열 수 없습니다')

sys.exit()

print('동영상 연결 성공!')

print('가로 사이즈: ', int(cap.get(cv2.CAP\_PROP\_FRAME\_WIDTH)))

print('세로 사이즈: ', int(cap.get(cv2.CAP\_PROP\_FRAME\_HEIGHT)))

print('프레임 수: ', int(cap.get(cv2.CAP\_PROP\_FRAME\_COUNT)))

print('FPS: ', cap.get(cv2.CAP\_PROP\_FPS))

while True:

ret, frame = cap.read()

if not ret:

break

cv2.imshow('frame', frame)

if cv2.waitKey(10) == 27: # ESC

break

cap.release()

## ##4. cameraout.py

import cv2

cap = cv2.VideoCapture(0)

w = round(cap.get(cv2.CAP\_PROP\_FRAME\_WIDTH))

h = round(cap.get(cv2.CAP\_PROP\_FRAME\_HEIGHT))

fps = cap.get(cv2.CAP\_PROP\_FPS)

fourcc = cv2.VideoWriter.fourcc(\*'DIVX')

out = cv2.VideoWriter('output.avi', fourcc, fps, (w, h))

while True:

ret, frame = cap.read()

if not ret:

break

out.write(frame)

cv2.imshow('frame', frame)

if cv2.waitKey(10) == 27:

break

cap.release()

out.release()

## ##5. videoOut.py

import cv2

# 두개의 동영상을 읽어들어와서 하나의 영상으로 출력해보기

# 비디오객체 1,2 생성

cap1 = cv2.VideoCapture('./tiger.mp4')

cap2 = cv2.VideoCapture('./clouds.mp4')

# 영상의 총 프레임수 반올림

frame\_cnt1 = round(cap1.get(cv2.CAP\_PROP\_FRAME\_COUNT))

frame\_cnt2 = round(cap2.get(cv2.CAP\_PROP\_FRAME\_COUNT))

print(frame\_cnt1)

print(frame\_cnt2)

# 1초당 프레임수 반환

fps1 = cap1.get(cv2.CAP\_PROP\_FPS)

fps2 = cap2.get(cv2.CAP\_PROP\_FPS)

print(fps1)

print(fps2)

# 영상의 가로, 세로

w = round(cap1.get(cv2.CAP\_PROP\_FRAME\_WIDTH))

h = round(cap1.get(cv2.CAP\_PROP\_FRAME\_HEIGHT))

print('width', w)

print('height', h)

# VideoWriter객체 생성 , 코드포맷 방식 : Divx

fourcc = cv2.VideoWriter.fourcc(\*'DIVX')

# 비디오 내보내기\_파일명 / 코드포맷 객체 / 초당 fps / 프레임사이즈 튜플

out = cv2.VideoWriter('mix.avi', fourcc, fps1, (w,h))

# 총 프레임수를 돌면서 out객체에 write메서드로 frame을 넣어주기(영상 혼합)

for i in range(frame\_cnt1):

ret, frame = cap1.read() # 영상 읽어오기

cv2.imshow('output', frame) # 출력

out.write(frame) # out객체에 frame 씌우기

if cv2.waitKey(10) == 27:

break

for i in range(frame\_cnt2):

ret, frame = cap2.read()

cv2.imshow('output', frame)

out.write(frame) # out객체에 frame 씌우기

if cv2.waitKey(10) == 27:

break

cap1.release()

cap2.release()

out.release()

## ## 책갈피

## ##6. keyevent.py

import cv2

img = cv2.imread('./dog.bmp')

cv2.imshow('image', img)

while True:

keyvalue = cv2.waitKey()

if keyvalue == ord('i') or keyvalue == ord('I'):

img = ~img

cv2.imshow('image',img)

elif keyvalue == 27: # ESC

break

## ##7. mouseEvent.py

import cv2

import numpy as np

oldx = oldy = 0

def onMouse(event, x, y, flags, param):

global oldx, oldy

# print(event)

if event == cv2.EVENT\_LBUTTONDOWN: # 왼쪽 버튼 눌릴 때

oldx, oldy = x, y

print('왼쪽 버튼이 눌렸어요: %d, %d' % (x,y))

elif event == cv2.EVENT\_LBUTTONUP:

print('왼쪽 버튼이 떼졌어요: %d, %d' % (x,y))

elif event == cv2.EVENT\_MOUSEMOVE:

if flags & cv2.EVENT\_FLAG\_LBUTTON:

cv2.line(img, (oldx,oldy), (x,y), (255,51,255), 3)

cv2.imshow('img', img)

oldx, oldy = x, y

img = np.ones((500,500,3), dtype=np.uint8) \* 255

cv2.namedWindow('img') # 창의 이름을 지정

cv2.setMouseCallback('img', onMouse)

cv2.imshow('img', img)

cv2.waitKey()

# #3. Day3

## ##1. add.py

import cv2

src1 = cv2.imread('./dog.bmp', cv2.IMREAD\_GRAYSCALE)

src2 = cv2.imread('./dog.bmp')

cv2.imshow('src1', src1)

cv2.imshow('src2', src2)

dst1 = cv2.add(src1, 100)

dst2 = cv2.add(src2, (100,100,100,0)) # 가장 뒤는 채널이라서 더하면 x

cv2.imshow('dst1', dst1)

cv2.imshow('dst2', dst2)

dst3 = cv2.subtract(src1, 100)

cv2.imshow('dst3',dst3)

dst4 = cv2.multiply(src1, 10)

cv2.imshow('dst4',dst4)

dst5 = cv2.divide(src1, 10)

cv2.imshow('dst5',dst5)

cv2.waitKey()

## ##2. blending.py

import cv2

import matplotlib.pyplot as plt

src1 = cv2.imread('./man.jpg')

src2 = cv2.imread('./turkey.jpg')

# src1 + src2 값이 255를 넘어가면 해당값의 256을 빼서 표현

dst1 = src1 + src2 # +: 특정 밝기를 넘어서면 색상반전

# cv2.add(src1, src2)의 값이 55를 넘어가면 255로 고정

dst2 = cv2.add(src1, src2) # add: 최대값이 흰색

img = {'src1': src1, 'src2':src2, 'dst1':dst1, 'dst2':dst2}

for i, (k,v) in enumerate(img.items()):

plt.subplot(2,2,i+1)

plt.imshow(v[:,:,::-1])

plt.title(k)

plt.show()

## ##3. addWeighted.py

import cv2

import matplotlib.pyplot as plt

src1 = cv2.imread('./man.jpg')

src2 = cv2.imread('./turkey.jpg')

# src1 + src2 값이 255를 넘어가면 해당값의 256을 빼서 표현

dst1 = src1 + src2 # +: 특정 밝기를 넘어서면 색상반전

# cv2.add(src1, src2)의 값이 55를 넘어가면 255로 고정

dst2 = cv2.add(src1, src2) # add: 최대값이 흰색

img = {'src1': src1, 'src2':src2, 'dst1':dst1, 'dst2':dst2}

for i, (k,v) in enumerate(img.items()):

plt.subplot(2,2,i+1)

plt.imshow(v[:,:,::-1])

plt.title(k)

plt.show()

## ##4. arithmetic.py

import cv2

from matplotlib import pyplot as plt

src1 = cv2.imread('./dog.jpg')

src2 = cv2.imread('./square.bmp')

alpha = 0.5

dst1 = cv2.add(src1, src2) # 최대값 255:흰색

dst2 = cv2.addWeighted(src1, alpha, src2, (1-alpha), 0) # 강아지50 : 흰색50

dst3 = cv2.subtract(src1, src2) # 최소값 0: 검은색

dst4 = cv2.absdiff(src1, src2) # -값을 +로 반전해서 값 추출

img = {'dst1':dst1, 'dst2':dst2,'dst3':dst3, 'dst4':dst4}

for i, (k,v) in enumerate(img.items()):

plt.subplot(2,2,i+1)

plt.imshow(v[:,:,::-1])

plt.title(k)

plt.show()

'''

cv2.imshow('dst1',dst1)

cv2.imshow('dst2',dst2)

cv2.imshow('dst3',dst3)

cv2.imshow('dst4',dst4)

cv2.waitKey()

'''

## ##5. color.py

import cv2

src = cv2.imread('./candies.png') # BGR

print('shape: ', src.shape)

print('dtype: ', src.dtype)

b, g, r = cv2.split(src) # 채널 분리

'''

# ndarray 인덱싱 활용 - 채널

b = src[:,:,0]

g = src[:,:,1]

r = src[:,:,2]

'''

cv2.imshow('src', src)

cv2.imshow('b', b) # 각 색상에 해당하는 부분이 흰색으로 보임

cv2.imshow('g', g)

cv2.imshow('r', r)

cv2.waitKey()

## ##6. hist1.py

import cv2

import matplotlib.pyplot as plt

src = cv2.imread('./dog.bmp', cv2.IMREAD\_GRAYSCALE)

hist = cv2.calcHist([src], [0], None, [256], [0,255])

# y축은 픽셀 / x축은 빈의갯수(0~255)

cv2.imshow('src',src)

plt.plot(hist)

plt.show()

cv2.waitKey()

## ##7. hist2.py

import cv2

import matplotlib.pyplot as plt

src = cv2.imread('./dog.bmp')

bgr = cv2.split(src)

colors = ['b', 'g', 'r']

for (b,c) in zip(bgr, colors):

hist = cv2.calcHist([b], [0], None, [256], [0,256])

plt.plot(hist, color=c)

cv2.imshow('src', src)

plt.show()

cv2.waitKey()

# #4. Day4

## ##1. equalize1.py

import cv2

import matplotlib.pyplot as plt

# 흑백사진 이퀄라이즈

src = cv2.imread('./Hawkes.jpg', cv2.IMREAD\_GRAYSCALE)

dst = cv2.equalizeHist(src) # 뭉쳐있는 정보를 펼쳐줌

hist1 = cv2.calcHist([src], [0], None, [256], [0, 255])

hist2 = cv2.calcHist([dst], [0], None, [256], [0, 255])

hists = {'hist1': hist1, 'hist2': hist2}

# cv2.imshow('src', src)

# cv2.imshow('dst', dst)

# cv2.waitKey()

plt.figure(figsize=(12, 8))

for i, (k, v) in enumerate(hists.items()):

plt.subplot(1, 2, i+1)

plt.title(k)

plt.plot(v)

plt.show()

## ##2. equalize2.py

import cv2

src = cv2.imread('./field.bmp')

ycbcr = []

'''

dst = cv2.cvtColor(src, cv2.COLOR\_BGR2YCrCb)

ycbcr = cv2.split(dst)

ycbcr = list(ycbcr)

# print(ycbcr)

# 밝기정보(y) 평탄화

ycbcr[0] = cv2.equalizeHist(ycbcr[0])

dst = cv2.merge(ycbcr)

dst = cv2.cvtColor(dst, cv2.COLOR\_YCrCb2BGR)

cv2.imshow('src',src)

cv2.imshow('dst',dst)

cv2.waitKey()

'''

# 문제

# split(), merge()를 사용하지 않고 슬라이싱 or 인덱싱을 써서 동일한 프로세스

dst = cv2.cvtColor(src, cv2.COLOR\_BGR2YCrCb)

dst[:,:,0] = cv2.equalizeHist(dst[:,:,0])

dst = cv2.cvtColor(dst, cv2.COLOR\_YCrCb2BGR)

cv2.imshow('src',src)

cv2.imshow('dst',dst)

cv2.waitKey()

## ##3. clahe.py

import cv2

img = cv2.imread('./field.bmp')

dst = cv2.cvtColor(img, cv2.COLOR\_BGR2YCrCb)

img\_eq = dst.copy()

img\_clahe = dst.copy()

img\_eq[:,:,0] = cv2.equalizeHist(img\_eq[:,:,0])

img\_eq = cv2.cvtColor(img\_eq, cv2.COLOR\_YCrCb2BGR)

clahe = cv2.createCLAHE(4,(4,4))

img\_clahe[:,:,0] = clahe.apply(img\_clahe[:,:,0])

img\_clahe = cv2.cvtColor(img\_clahe, cv2.COLOR\_YCrCb2BGR)

cv2.imshow('img', img)

cv2.imshow('img\_eq', img\_eq)

cv2.imshow('img\_clahe', img\_clahe)

cv2.waitKey()

## ##4. normalize.py

import cv2

import numpy as np

import matplotlib.pyplot as plt

img = cv2.imread('./Hawkes.jpg', cv2.IMREAD\_GRAYSCALE)

img\_norm1 = img.astype(np.float32)

img\_norm1 = ((img\_norm1 - img\_norm1.min()) \* 255 / (img\_norm1.max() - img\_norm1.min()))

img\_norm1 = img\_norm1.astype(np.uint8)

img\_norm2 = cv2.normalize(img, None, 0 , 255, cv2.NORM\_MINMAX)

# cv2.imshow('img', img)

# cv2.imshow('img\_norm1', img\_norm1)

# cv2.imshow('img\_norm2', img\_norm2)

# cv2.waitKey()

hist = cv2.calcHist([img], [0], None, [256], [0, 255])

hist\_norm1 = cv2.calcHist([img\_norm1], [0], None, [256], [0, 255])

hist\_norm2 = cv2.calcHist([img\_norm2], [0], None, [256], [0, 255])

hists = {'hist': hist, 'hist\_norm1': hist\_norm1, 'hist\_norm2': hist\_norm2}

for i, (k, v) in enumerate(hists.items()):

plt.subplot(1, 3, i+1)

plt.title(k)

plt.plot(v)

plt.show()

## ##5. inRange.py

import cv2

src = cv2.imread('./candies.png')

hsv = cv2.cvtColor(src, cv2.COLOR\_BGR2HSV)

'''50 <= H <= 80

150 <= S <= 255

0 <= V <= 255

'''

dst = cv2.inRange(hsv, (50,150,0), (80,255,255))

cv2.imshow('src',src)

cv2.imshow('dst',dst)

cv2.waitKey()

## ##6. copyTo.py

import cv2

src = cv2.imread('./airplane.bmp')

mask = cv2.imread('./mask\_plane.bmp')

dst = cv2.imread('./field.bmp')

cv2.copyTo(src, mask, dst)

cv2.imshow('src',src)

cv2.imshow('dst',dst)

cv2.waitKey()

## ##7. chromakey.py

import cv2

cap1 = cv2.VideoCapture('./woman.mp4')

cap2 = cv2.VideoCapture('./ham.mp4')

w = round(cap1.get(cv2.CAP\_PROP\_FRAME\_WIDTH))

h = round(cap1.get(cv2.CAP\_PROP\_FRAME\_HEIGHT))

frame\_cnt1 = round(cap1.get(cv2.CAP\_PROP\_FRAME\_COUNT))

frame\_cnt2 = round(cap2.get(cv2.CAP\_PROP\_FRAME\_COUNT))

fps = round(cap1.get(cv2.CAP\_PROP\_FPS))

print(w)

print(h)

print(frame\_cnt1)

print(frame\_cnt2)

print(fps)

isKeypress = False

while True:

ret1, frame1 = cap1.read()

if not ret1:

break

if isKeypress:

ret2, frame2 = cap2.read()

if not ret2:

break

hsv = cv2.cvtColor(frame1, cv2.COLOR\_BGR2HSV)

mask = cv2.inRange(hsv, (50, 150, 0), (70, 255, 255))

cv2.copyTo(frame2, mask, frame1)

cv2.imshow('frame1', frame1)

key = cv2.waitKey(10)

if key == ord(' '):

isKeypress = not isKeypress

elif key == 27:

break

cap1.release()

cap2.release()

## ##8. roi.py

import cv2

img1 = cv2.imread('./sun.jpg')

# x, y, w, h

x = 182

y = 21

w = 122

h = 110

roi = img1[y:y+h,x:x+w]

img2 = roi.copy()

img1[y:y+h, x+w:x+w+w] = roi

cv2.rectangle(img1, (x,y), (x+w+w, y+h), (0,255,0), 3)

cv2.imshow('img1', img1)

cv2.imshow('roi', img2)

cv2.waitKey()

## 

## ##9. selectROI1.py

import cv2

img = cv2.imread('./sun.jpg')

x, y, w, h = cv2.selectROI('img', img, False)

if w and h:

roi = img[y:y+h, x: x+w]

cv2.imshow('roi', roi)

cv2.waitKey()

## ##10. selectROI2\_result.py

import cv2

isDrag = False

oldx = oldy = w = h = 0

color = (255,0,0)

img\_copy = None

def on\_mouse(event, x, y, flags, param):

global oldx, oldy, w, h , isDrag, img\_copy

if event == cv2.EVENT\_LBUTTONDOWN:

isDrag = True

oldx = x

oldy = y

elif event == cv2.EVENT\_MOUSEMOVE:

if isDrag:

img\_copy = img.copy()

cv2.rectangle(img\_copy, (oldx,oldy), (x,y), color, 3)

cv2.imshow('img', img\_copy)

elif event == cv2.EVENT\_LBUTTONUP:

if isDrag:

isDrag = False

if x > oldx and y > oldy:

w = x - oldx

h = y - oldy

if w > 0 and h > 0:

cv2.rectangle(img\_copy, (oldx,oldy), (x,y), color, 3)

roi = img[oldy:oldy+h, oldx:oldx+w]

cv2.imshow('roi', roi)

else:

cv2.imshow('img', img)

print('영역이 잘못되었음')

img = cv2.imread('./sun.jpg')

cv2.imshow('img',img)

cv2.setMouseCallback('img', on\_mouse)

cv2.waitKey()

# 이벤트 발생하면

# #5. Day5

## ##1. binaryzation.py

import cv2

import matplotlib.pyplot as plt

src = cv2.imread('./cells.png', cv2.IMREAD\_GRAYSCALE)

hist = cv2.calcHist([src], [0], None, [256], [0,255])

# 임계값보다 낮으면 검정 / 높으면 흰색(\_: 설정한 thresh 값)

\_, dst1 = cv2.threshold(src, 100, 255, cv2.THRESH\_BINARY) # 임계값 / 결과물

\_, dst2 = cv2.threshold(src, 210, 255, cv2.THRESH\_BINARY)

cv2.imshow('src',src)

cv2.imshow('dst1',dst1)

cv2.imshow('dst2',dst2)

plt.plot(hist)

plt.show()

cv2.waitKey()

## ##2. otsu.py

import cv2

src = cv2.imread('./rice.png', cv2.IMREAD\_GRAYSCALE)

th, dst = cv2.threshold(src, 0, 255, cv2.THRESH\_BINARY | cv2.THRESH\_OTSU)

print(th)

cv2.imshow('src', src)

cv2.imshow('dst', dst)

cv2.waitKey()

## ##3. local.py

import cv2

import numpy as np

src = cv2.imread('./rice.png', cv2.IMREAD\_GRAYSCALE)

# 전역 이진화

\_, dst1 = cv2.threshold(src, 0, 255, cv2.THRESH\_BINARY | cv2.THRESH\_OTSU)

# 지역 이진화

dst2 = np.zeros(src.shape, np.uint8)

bw = src.shape[1] // 4

bh = src.shape[0] // 4

for y in range(4):

for x in range(4):

src\_ = src[y\*bh: (y+1)\*bh, x\*bw: (x+1)\*bw]

dst\_ = dst2[y\*bh: (y+1)\*bh, x\*bw: (x+1)\*bw]

cv2.threshold(src\_, 0, 255, cv2.THRESH\_BINARY | cv2.THRESH\_OTSU, dst\_)

cv2.imshow('src', src)

cv2.imshow('dst1', dst1)

cv2.imshow('dst2', dst2)

cv2.waitKey()

## ##4. adaptive.py

import cv2

import matplotlib.pyplot as plt

img = cv2.imread('./sudoku.jpg', cv2.IMREAD\_GRAYSCALE)

th, dst1 = cv2.threshold(img, 0, 255, cv2.THRESH\_BINARY | cv2.THRESH\_OTSU)

dst2 = cv2.adaptiveThreshold(img, 255, cv2.ADAPTIVE\_THRESH\_MEAN\_C, cv2.THRESH\_BINARY, 9, 5)

dst3 = cv2.adaptiveThreshold(img, 255, cv2.ADAPTIVE\_THRESH\_GAUSSIAN\_C, cv2.THRESH\_BINARY, 9, 5)

dic = {'img':img, 'dst1':dst1, 'dst2':dst2, 'dst3':dst3}

for i, (k,v) in enumerate(dic.items()):

plt.subplot(2,2,i+1)

plt.title(k)

plt.imshow(v, 'gray')

plt.show()

## ##5. compare.py

import cv2

import matplotlib.pyplot as plt

import numpy as np

img1 = cv2.imread('./tae1.jpg')

img2 = cv2.imread('./tae2.jpg')

img3 = cv2.imread('./tae3.jpg')

img4 = cv2.imread('./dr1.jpg')

imgs = [img1,img2,img3,img4]

hists = []

for i, img in enumerate(imgs):

plt.subplot(1, len(imgs), i+1)

plt.title('img%d' % (i+1))

plt.axis('off')

plt.imshow(img[:,:,::-1])

hsv = cv2.cvtColor(img, cv2.COLOR\_BGR2HSV)

hist = cv2.calcHist([hsv], [0,1], None, [180,256], [0,180,0,256])

cv2.normalize(hist, hist, 0, 1, cv2.NORM\_MINMAX)

hists.append(hist)

query = hists[0]

methods = {'CORREL': cv2.HISTCMP\_CORREL, 'CHISQR': cv2.HISTCMP\_CHISQR,

'INTERSECT': cv2.HISTCMP\_INTERSECT, 'BHATTACHARYYA':cv2.HISTCMP\_BHATTACHARYYA}

for j, (name, flag) in enumerate(methods.items()):

print('%-10s' % name, end='\t')

for i, (hist, img) in enumerate(zip(hists, imgs)):

ret = cv2.compareHist(query, hist, flag)

if flag == cv2.HISTCMP\_INTERSECT:

ret = ret/np.sum(query)

print('img%d:%7.2f' % (i+1, ret), end='\t')

print()

plt.show()

## ##6. translate.py

import cv2

import numpy as np

img = cv2.imread('./dog.bmp')

# [1,0,a],[0,1,b] - 이동 변환행렬

aff = np.array([[1,0,150],[0,1,100]], dtype=np.float32)

dst = cv2.warpAffine(img, aff, (0,0))

cv2.imshow('src', img)

cv2.imshow('dst',dst)

cv2.waitKey()

## ##7. resize.py

import cv2

img = cv2.imread('./dog.bmp')

dst1 = cv2.resize(img, (1280, 1023), interpolation=cv2.INTER\_NEAREST) # 속도는 빠르지만 퀄리티 낮음

dst2 = cv2.resize(img, (1280, 1023), interpolation=cv2.INTER\_CUBIC)

cv2.imshow('img', img)

# cv2.imshow('dst1', dst1)

# cv2.imshow('dst2', dst2)

cv2.imshow('dst1', dst1[400:800,200:600]) # 일부추출

cv2.imshow('dst2', dst2[400:800,200:600])

cv2.waitKey()

## ##8\_rotation.py

import cv2

img = cv2.imread('./dog.bmp')

cp = (img.shape[1]/2, img.shape[0]/2) # 중심좌표 구하기

rot = cv2.getRotationMatrix2D(cp, 30, 0.5)

dst = cv2.warpAffine(img, rot, (0,0))

cv2.imshow('img', img)

cv2.imshow('dst', dst)

cv2.waitKey()

## ##9\_perspective.py

import cv2

import numpy as np

# 누워있는 사진을 똑바로 펼침

img = cv2.imread('./pic.jpg')

w, h = 600,400

srcQuad = np.array([[370,173],[1220,155],[1420,840],[210,850]], np.float32)

dstQuad = np.array([[0,0],[w,0],[w,h],[0,h]], np.float32)

pers = cv2.getPerspectiveTransform(srcQuad, dstQuad)

dst = cv2.warpPerspective(img, pers, (w,h))

cv2.imshow('img',img)

cv2.imshow('dst',dst)

cv2.waitKey()

## ##10\_namecard.py

import cv2

import numpy as np

import sys

src = cv2.imread('./namecard.jpg')

h, w = src.shape[:2]

dh = 500

# A4용지 크기: 210\*297cm

dw = round(dh \* 297 / 210)

srcQuad = np.array([[30, 30], [30, h-30], [w-30, h-30], [w-30, 30]], np.float32)

dstQuad = np.array([[0, 0], [0, dh], [dw, dh], [dw, 0]], np.float32)

dragSrc = [False, False, False, False]

def drawROI(img, corners):

cpy = img.copy()

c1 = (192, 192, 255)

c2 = (128, 128, 255)

for pt in corners:

cv2.circle(cpy, tuple(pt.astype(int)), 25, c1, -1)

cv2.line(cpy, tuple(corners[0].astype(int)), tuple(corners[1].astype(int)), c2, 2)

cv2.line(cpy, tuple(corners[1].astype(int)), tuple(corners[2].astype(int)), c2, 2)

cv2.line(cpy, tuple(corners[2].astype(int)), tuple(corners[3].astype(int)), c2, 2)

cv2.line(cpy, tuple(corners[3].astype(int)), tuple(corners[0].astype(int)), c2, 2)

return cpy

def onMouse(event, x, y, flags, param):

global srcQuad, dragSrc, ptOld, src

if event == cv2.EVENT\_LBUTTONDOWN:

for i in range(4):

if cv2.norm(srcQuad[i] - (x, y)) < 25:

dragSrc[i] = True

ptOld = (x, y)

break

if event == cv2.EVENT\_LBUTTONUP:

for i in range(4):

dragSrc[i] = False

if event == cv2.EVENT\_MOUSEMOVE:

for i in range(4):

if dragSrc[i]:

dx = x - ptOld[0]

dy = y - ptOld[1]

srcQuad[i] += (dx, dy)

cpy = drawROI(src, srcQuad)

cv2.imshow('img', cpy)

ptOld = (x, y)

break

disp = drawROI(src, srcQuad)

cv2.imshow('img', disp)

cv2.setMouseCallback('img', onMouse)

while True:

key = cv2.waitKey()

if key == 13:

break

elif key == 27:

sys.exit()

pers = cv2.getPerspectiveTransform(srcQuad, dstQuad)

dst = cv2.warpPerspective(src, pers, (dw,dh), flags=cv2.INTER\_CUBIC)

cv2.imshow('dst',dst)

cv2.waitKey()

# #6. Day6

## ##1\_blurring.py

import cv2

import matplotlib.pyplot as plt

import numpy as np

img = cv2.imread('./dog.bmp')

dst1 = cv2.cvtColor(img, cv2.COLOR\_BGR2RGB)

dst2 = cv2.blur(img, (3,3))

cv2.imshow('img', img)

cv2.imshow('dst2', dst2)

cv2.waitKey()

plt.figure(figsize=(10,5))

for i, k in enumerate([5,7,9]): # 커널숫자가 커질수록 더 뿌얘짐

kernel = np.ones((k, k)) / k \*\* 2

filtering = cv2.filter2D(dst1, -1, kernel)

plt.subplot(1,3,i+1)

plt.imshow(filtering)

plt.title('kernel size:{}'.format(k))

plt.axis('off')

plt.show()

## ##2\_gaussian.py

import cv2

img = cv2.imread('./dog.bmp', cv2.IMREAD\_GRAYSCALE)

dst1 = cv2.GaussianBlur(img, (0,0), 1)

dst2 = cv2.blur(img, (3,3))

cv2.imshow('img',img)

cv2.imshow('dst1',dst1)

cv2.imshow('dst2',dst2)

cv2.waitKey()

## ##3\_median.py

import cv2

img = cv2.imread('./noise.bmp', cv2.IMREAD\_GRAYSCALE)

dst = cv2.medianBlur(img, 3)

cv2.imshow('img', img)

cv2.imshow('dst', dst)

cv2.waitKey()

## ##4\_bilateral.py

import cv2

img = cv2.imread('./gaussian\_noise.jpg', cv2.IMREAD\_GRAYSCALE)

dst1 = cv2.GaussianBlur(img, (5,5),1)

dst2 = cv2.bilateralFilter(img, 5, 80, 80)

cv2.imshow('img',img)

cv2.imshow('dst1',dst1)

cv2.imshow('dst2',dst2)

cv2.waitKey()

## ##5\_Canny.py

import cv2

import numpy as np

img = cv2.imread('./dog.bmp')

med\_val = np.median(img)

lower = int(max(0, 0.7\*med\_val))

upper = int(min(255, 1.3\*med\_val))

print(med\_val)

print(lower)

print(upper)

dst = cv2.GaussianBlur(img, (3,3), 0, 0)

dst = cv2.Canny(dst, lower, upper, 3) # 낮출수록 잘게들어간 선 / 높일수록 큼직하게 선

cv2.imshow('img',img)

cv2.imshow('dst',dst)

cv2.waitKey()

## ##6\_filterCam.py

import cv2

import numpy as np

cap = cv2.VideoCapture(0)

#함수생성

#가우시안 함수

def blur\_filter(img):

img = cv2.GaussianBlur(img, (0,0), 3)

return img

# 캐니적용 함수

def canny\_filter(img):

med\_val = np.median(img)

lower = int(max(0, 0.5\*med\_val))

upper = int(min(255, 1.5 \* med\_val))

dst = cv2.GaussianBlur(img, (3,3), 0, 0)

dst = cv2.Canny(dst, lower, upper, 3)

return dst

cam\_mode = 0

while True:

ret, frame = cap.read()

if cam\_mode == 1: # 가우시안

frame = blur\_filter(frame)

elif cam\_mode == 2:

frame = canny\_filter(frame)

cv2.imshow('frame', frame)

key = cv2.waitKey(10)

if key == 27:

break

elif key == ord(' '):

cam\_mode += 1

if cam\_mode == 3:

cam\_mode = 0

cap.release()

# #7. Day7

## ##1\_morphology.py

import cv2

import numpy as np

# # kernel = cv2.getStructuringElement(cv2.MORPH\_RECT, (3,3))

# kernel = cv2.getStructuringElement(cv2.MORPH\_ELLIPSE, (5,5))

# print(kernel)

# 서킷이미지\_흑백으로

img = cv2.imread('./circuit.bmp', cv2.IMREAD\_GRAYSCALE)

# 모폴로치\_사각형 생성\_반드시 정방행렬은 아님

gse = cv2.getStructuringElement(cv2.MORPH\_RECT, (5,3)) # 무조건 정방행렬X

# 모폴로지처리\_침식+사각형

dst1 = cv2.erode(img,gse)

# 모폴로지처리\_팽창+사각형

dst2 = cv2.dilate(img,gse)

cv2.imshow('img', img)

cv2.imshow('dst1', dst1)

cv2.imshow('dst2', dst2)

cv2.waitKey()

# 결과\_ 침식: 선이 얇아지고 배경이 많아짐, 팽창: 선이 두꺼워지고 내부 구멍이 작아짐

## ##2\_labeling.py

import cv2

img = cv2.imread('./keyboard.bmp', cv2.IMREAD\_GRAYSCALE)

# 오츠이진화

\_, img\_bin = cv2.threshold(img,0,255, cv2.THRESH\_BINARY | cv2.THRESH\_OTSU)

# 색상변경: 그레이스케일 -> BGR

dst = cv2.cvtColor(img, cv2.COLOR\_GRAY2BGR)

cnt, labels, stats, centroids = cv2.connectedComponentsWithStats(img\_bin)

print("cnt: ",cnt) # 객체 갯수

print("labels: ",labels) # 레이블맵 행렬

print("stats: ",stats) # 크기: 폭, 높이, 면적 등 상태값

print("centroids: ",centroids) # 무게중심좌표

#객체 그려보기

for i in range(1, cnt):

(x,y,w,h,area) = stats[i]

if area < 30:

continue

cv2.rectangle(dst, (x,y,w,h), (0,255,255))

cv2.imshow('img', img)

cv2.imshow('img\_bin', img\_bin)

cv2.imshow('dst', dst)

cv2.waitKey()

## ##3\_contours1.py

import cv2

import random

img = cv2.imread('./contours.bmp', cv2.IMREAD\_GRAYSCALE)

# 외곽선 끝점의 좌표만 반환

contours, \_ = cv2.findContours(img, cv2.RETR\_CCOMP, cv2.CHAIN\_APPROX\_NONE)

# print(contours)

# 색상변환\_그레이스케일->BGR

dst = cv2.cvtColor(img, cv2.COLOR\_GRAY2BGR)

# 랜덤 색상 생성

color = (random.randint(0,255), random.randint(0,255),random.randint(0,255))

print(color)

# 외곽선 그리기\_이미지:dst, 좌표:contours, 인덱스:-1(모두), 색상, 두께:3

cv2.drawContours(dst, contours, -1, color,3)

cv2.imshow('img', img)

cv2.imshow('dst',dst)

cv2.waitKey()

# 결과: 잡히는 모든 객체에 랜덤색상의 외곽선그림

## ##4\_contours2.py

import cv2

import random

import numpy as np

img = cv2.imread('./milkdrop.bmp', cv2.IMREAD\_GRAYSCALE)

# 오츠이진화

\_, img\_bin = cv2.threshold(img, 0, 255, cv2.THRESH\_BINARY | cv2.THRESH\_OTSU)

contours, \_ = cv2.findContours(img\_bin, cv2.RETR\_CCOMP, cv2.CHAIN\_APPROX\_NONE)

h,w = img.shape[:2]

dst = np.zeros((h,w,3), np.uint8)

for i in range(len(contours)):

color = (random.randint(0,255),random.randint(0,255),random.randint(0,255))

cv2.drawContours(dst, contours, i, color, 2)

cv2.imshow('img', img)

cv2.imshow('img\_bin',img\_bin)

cv2.imshow('dst',dst)

cv2.waitKey()

## ##5\_contour3.py

import cv2

import math

def setLabel(img, pts, label):

(x,y,w,h) = cv2.boundingRect(pts)

pt1 = (x,y)

pt2 = (x+w, y+h)

cv2.rectangle(img, pt1, pt2, (0,0,255),2)

cv2.putText(img,label,pt1,cv2.FONT\_HERSHEY\_SIMPLEX,1,(0,0,255))

img = cv2.imread('./polygon.bmp')

gray = cv2.cvtColor(img, cv2.COLOR\_BGR2GRAY)

# 오츠이진화\_배경이 흰색으로 \_INV사용

\_, img\_bin = cv2.threshold(gray,0,255,cv2.THRESH\_BINARY\_INV | cv2.THRESH\_OTSU)

# 외곽선 추출

contours, \_ = cv2.findContours(img\_bin, cv2.RETR\_EXTERNAL, cv2.CHAIN\_APPROX\_NONE)

for pts in contours:

if cv2.contourArea(pts) < 200:

continue

approx = cv2.approxPolyDP(pts, cv2.arcLength(pts, True) \* 0.02, True)

# print(approx)

vtc = len(approx)

print(vtc)

if vtc == 3:

setLabel(img, pts, 'TRI')

elif vtc == 4:

setLabel(img, pts, 'RECT')

else:

length = cv2.arcLength(pts, True)

area = cv2.contourArea(pts)

ratio = 4. \* math.pi \* area / (length \* length) # 원인지 판단공식

if ratio > 0.70:

setLabel(img, pts, 'CIR')

else:

setLabel(img, pts, 'NONAME')

cv2.imshow('img',img)

cv2.waitKey()

## ##6\_tesseract.py

import cv2

import pytesseract

img = cv2.imread('./hello.png')

# 색상변경 RGB\_테서렉트에서는 RGB로 인식

dst = cv2.cvtColor(img,cv2.COLOR\_BGR2RGB)

# 이미지를 스트링으로\_lang=kor(한국어)+eng(영어)

text = pytesseract.image\_to\_string(dst, lang='kor+eng')

print(text)

## ##7\_namecard\_yys.py

import cv2

import numpy as np

import sys

import pytesseract

src = cv2.imread('namecard.jpg')

h, w = src.shape[:2]

dh = 500

# A4용지 크기: 210\*297cm

dw = round(dh \* 297 / 210)

srcQuad = np.array([[30, 30], [30, h-30], [w-30, h-30], [w-30, 30]], np.float32)

dstQuad = np.array([[0, 0], [0, dh], [dw, dh], [dw, 0]], np.float32)

dragSrc = [False, False, False, False]

def drawROI(img, corners):

cpy = img.copy()

c1 = (192, 192, 255)

c2 = (128, 128, 255)

for pt in corners:

cv2.circle(cpy, tuple(pt.astype(int)), 25, c1, -1)

cv2.line(cpy, tuple(corners[0].astype(int)), tuple(corners[1].astype(int)), c2, 2)

cv2.line(cpy, tuple(corners[1].astype(int)), tuple(corners[2].astype(int)), c2, 2)

cv2.line(cpy, tuple(corners[2].astype(int)), tuple(corners[3].astype(int)), c2, 2)

cv2.line(cpy, tuple(corners[3].astype(int)), tuple(corners[0].astype(int)), c2, 2)

return cpy

def onMouse(event, x, y, flags, param):

global srcQuad, dragSrc, ptOld, src

if event == cv2.EVENT\_LBUTTONDOWN:

for i in range(4):

if cv2.norm(srcQuad[i] - (x, y)) < 25:

dragSrc[i] = True

ptOld = (x, y)

break

if event == cv2.EVENT\_LBUTTONUP:

for i in range(4):

dragSrc[i] = False

if event == cv2.EVENT\_MOUSEMOVE:

for i in range(4):

if dragSrc[i]:

dx = x - ptOld[0]

dy = y - ptOld[1]

srcQuad[i] += (dx, dy)

cpy = drawROI(src, srcQuad)

cv2.imshow('img', cpy)

ptOld = (x, y)

break

disp = drawROI(src, srcQuad)

cv2.imshow('img', disp)

cv2.setMouseCallback('img', onMouse)

while True:

key = cv2.waitKey()

if key == 13:

break

elif key == 27:

sys.exit()

pers = cv2.getPerspectiveTransform(srcQuad, dstQuad)

dst = cv2.warpPerspective(src, pers, (dw,dh), flags=cv2.INTER\_CUBIC)

# 오츠이진화를 위한 그레이스케일

dst = cv2.cvtColor(dst, cv2.COLOR\_RGB2GRAY)

# 팽창연산\_너무과하게돼서 커널사이즈 최대한 작게\_사각형

gse = cv2.getStructuringElement(cv2.MORPH\_RECT, (1,1))

dst = cv2.erode(dst,gse)

# 오츠 이진화

\_, dst = cv2.threshold(dst, 0, 255, cv2.THRESH\_BINARY\_INV | cv2.THRESH\_OTSU)

text = pytesseract.image\_to\_string(dst, lang='kor+eng')

print(text)

cv2.imshow('dst',dst)

cv2.waitKey()

## ##7\_namecard.py

import cv2

import pytesseract

import numpy as np

def reorderPts(pts):

# print(pts)

'''

[[903. 199.]

[179. 200.]

[159. 593.]

[938. 581.]]

[[179. 200.]

[159. 593.]

[938. 581.]

[903. 199.]]

'''

# print(pts[:, 1])

# print(pts[:, 0])

idx = np.lexsort((pts[:, 1], pts[:, 0]))

pts = pts[idx]

# print(idx)

if pts[0, 1] > pts[1, 1]:

pts[[0, 1]] = pts[[1, 0]]

if pts[2, 1] < pts[3, 1]:

pts[[2, 3]] = pts[[3, 2]]

print(pts)

return pts

src = cv2.imread('./namecard.jpg')

dw, dh = 700, 400

srcQuad = np.array([[0, 0], [0, 0], [0, 0], [0, 0]], np.float32)

dstQuad = np.array([[0, 0], [0, dh], [dw, dh], [dw, 0]], np.float32)

dst = np.zeros((dh, dw), np.uint8)

src\_gray = cv2.cvtColor(src, cv2.COLOR\_BGR2GRAY)

\_, src\_bin = cv2.threshold(src\_gray, 0, 255, cv2.THRESH\_BINARY | cv2.THRESH\_OTSU)

contours, \_ = cv2.findContours(src\_bin, cv2.RETR\_EXTERNAL, cv2.CHAIN\_APPROX\_NONE)

cpy = src.copy()

for pts in contours:

if cv2.contourArea(pts) < 1000:

continue

approx = cv2.approxPolyDP(pts, cv2.arcLength(pts, True) \* 0.02, True)

# print(approx)

# 영상에 다각형을 그림

# polylines(영상, 꼭지점 좌표, 폐곡선 여부, 선색상, 선두께)

cv2.polylines(cpy, [approx], True, (0, 255, 0), 2)

srcQuad = reorderPts(approx.reshape(4, 2).astype(np.float32))

pers = cv2.getPerspectiveTransform(srcQuad, dstQuad)

dst = cv2.warpPerspective(src, pers, (dw, dh))

dst\_gray = cv2.cvtColor(dst, cv2.COLOR\_BGR2GRAY)

print(pytesseract.image\_to\_string(dst\_gray, lang='kor+eng'))

cv2.imshow('src', src)

cv2.imshow('dst', dst)

cv2.waitKey()

## ## 7-1\_lexsort.py

import numpy as np

ndarray1 = np.array([1, 5, 1, 4, 3])

ndarray2 = np.array([9, 4, 0, 4, 0])

# 0, 0, 4, 4, 9 -> 2, 0, 4, 3, 1

# ndarray2 배열을 기준으로 오름차순으로 정렬하고 동일한 값의 경우

# ndarray1 배열의 기준으로 오름차순 한 인덱스를 반환

result = np.lexsort((ndarray2, ndarray1))

print(result)

# ndarray1 정렬 : 1,1,3,4,5 -> 겹치는 1중 ndarray2순서로는 2가 더 큼

# 결과 2,0,4,3,1