

# OpenCV-Python 기초사용법

## 영상속성

```
In [1]: import sys
import cv2
import numpy as np
```

```
In [2]: # cv2.imread(filename[, flags]) -> retval

# retval
# numpy.ndarray: retval.ndim/shape/size/dtype
# dtype: uint8
# shape: gray영상의 경우 (h,w) 또는 color (h,w, 3)
# gray영상 : cv2.CV_8UC1 -> numpy.uint8
# color영상 : cv2.CV_8UC3 -> numpy.uint8
```

```
In [3]: # 영상 불러오기
img1 = cv2.imread('fig/puppy.bmp', cv2.IMREAD_GRAYSCALE)
img2 = cv2.imread('fig/puppy_1280_853.jpg', cv2.IMREAD_COLOR)

if img1 is None or img2 is None:
    print('Image load failed!')
    sys.exit()

# 영상의 속성 참조
print('type(img1):', type(img1))
print('img1.shape:', img1.shape)
print('img2.shape:', img2.shape)
print('img1.dtype:', img1.dtype)
print('img2.dtype:', img2.dtype)

print('img1.shape length:', len(img1.shape))
print('img2.shape length:', len(img2.shape))
```

```
type(img1): <class 'numpy.ndarray'>
img1.shape: (480, 640)
img2.shape: (853, 1280, 3)
img1.dtype: uint8
img2.dtype: uint8
img1.shape length: 2
img2.shape length: 3
```

## 영상의 크기 참조

```
In [4]: h, w = img1.shape
print('img1 size: {} x {}'.format(w, h))

h, w = img2.shape[:2]
print('img2 size: {} x {}'.format(w, h))
```

```
img1 size: 640 x 480
img2 size: 1280 x 853
```

## 영상의 픽셀값 참조

In [7]:

```
x = 230
y = 320

p1 =img1[y,x]
print(p1)

p2 = img2[y, x]
print(p2)

### 픽셀값 바꾸기
img1[10:200, 10:200] = 0
img2[10:200, 10:200] = (0, 0, 255)

cv2.imshow('image', img1)
cv2.imshow('image2',img2)

cv2.waitKey()
cv2.destroyAllWindows()
```

```
128
[210 216 227]
```

## 영상생성

In [9]:

```
import numpy as np
import cv2

# 새 영상 생성하기
# img1 = np.empty((240, 320), dtype=np.uint8) # grayscale image
img1 = np.random.randint(0, 255, (240, 320), dtype = np.uint8) # gray random
img2 = np.zeros((240, 320, 3), dtype=np.uint8) # color image
img3 = np.ones((240, 320), dtype=np.uint8) * 255 # dark gray
img4 = np.full((240, 320, 3), (0, 255, 255), dtype=np.uint8) # yellow

cv2.imshow('img1', img1)
cv2.imshow('img2', img2)
cv2.imshow('img3', img3)
cv2.imshow('img4', img4)

cv2.waitKey()
cv2.destroyAllWindows()
```

## 새영상 생성

In [30]:

```
# 영상 복사
img1 = cv2.imread('fig/puppy.bmp', cv2.IMREAD_COLOR)
# img1 = cv2.imread('HappyFish.jpg')

if img1 is None:
    print("image load failed")
    sys.exit()

img2 = img1
# img2 = img1[150:250, 200:500]

img3 = img1.copy()

# img2[:, :] = (0, 0, 255)
```

```
img1[200:300,240:400] = (0, 255, 255)

print(img1.shape)
print(img1.dtype)

cv2.imshow('img1', img1)
cv2.imshow('img2', img2)
cv2.imshow('img3', img3)

cv2.waitKey()
cv2.destroyAllWindows()
```

```
-----
error                                     Traceback (most recent call last)
<ipython-input-30-e3958c80ceda> in <module>
    11
    12 img3 = img1.copy()
----> 13 img4 = cv2.copyTo(img1)
    14 # img2[:] = (0, 0, 255)
    15

error: OpenCV(4.5.4-dev) :-1: error: (-5:Bad argument) in function 'copyTo'
> Overload resolution failed:
> - copyTo() missing required argument 'mask' (pos 2)
> - copyTo() missing required argument 'mask' (pos 2)
```

## 부분 영상 추출

In [28]:

```
img1 = cv2.imread('fig/puppy.bmp')

img2 = img1[200:400, 300:500] # numpy.ndarray의 슬라이싱
img3 = img1[200:400, 300:500].copy()

# img1.fill(255)
cv2.circle(img2, (100, 100), 50, (0, 0, 255), 3)

cv2.imshow('img1', img1)
cv2.imshow('img2', img2)
cv2.imshow('img3', img3)

cv2.waitKey()
cv2.destroyAllWindows()
```

## 마스크 연산과 ROI

In [33]:

```
# 마스크 영상을 이용한 영상 합성
# cv2.copyTo(src, mask, dst = None) -> dst

src = cv2.imread('fig/airplane.bmp', cv2.IMREAD_COLOR)
mask = cv2.imread('fig/mask_plane.bmp', cv2.IMREAD_GRAYSCALE)
dst = cv2.imread('fig/field.bmp', cv2.IMREAD_COLOR)

if src is None or mask is None or dst is None:
    print('Image read failed!')
    sys.exit()

# 영상의 포맷과 형식이 같아야 함
cv2.copyTo(src, mask, dst)
```

```
# dst = cv2.copyTo(src, mask)

# Using numpy
# dst[mask > 0] = src[mask > 0]

cv2.imshow('src', src)
cv2.imshow('dst', dst)
cv2.imshow('mask', mask)

cv2.waitKey()
cv2.destroyAllWindows()
```

## 알파 채널을 마스크 영상으로 이용

In [50]:

```
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))

if src is None or sunglass is None:
    print('Image read failed!')
    sys.exit()

# 0(완전 투명) ~ 255(완전 불투명)
mask = sunglass[:, :, -1] # mask는 알파 채널로 만든 마스크 영상
glass = sunglass[:, :, 0:3] # glass는 b, g, r 3채널로 구성된 컬러 영상

h, w = mask.shape[:2]
crop = src[120:120+h, 220:220+w] # glass mask와 같은 크기의 부분 영상 추출

cv2.copyTo(glass, mask, crop)
# crop[mask > 0] = (0, 0, 255)

cv2.imshow('src', src)
cv2.imshow('glass', glass)
cv2.imshow('mask', mask)
# cv2.imshow('crop', crop)

# cv2.imwrite('puppy_sunglass.bmp', src)

cv2.waitKey()
cv2.destroyAllWindows()
```

## OpenCV 그리기 함수

In [11]:

```
# cv2.line(img, pt1, pt2, color, thickness = None, lineType = None, shift = None)
# flags
# img:그림을 그릴 영상
# pt1, pt2: 직선의 시작점, 끝점
# color: 직선의 칼라 (B,G,R)의 튜플
# thickness: 선두께, 기본은 1
# lineType: cv2.LINE_4, cv2.LINE_8, cv2.LINE_AA

# cv2.rectangle(img, pt1, pt2, color, thickness = None, lineType = None) -> img
# pt1 :좌측 상단, pt2: 우측하단
# cv2.rectangle(img, rect, color, thickness = None, lineType = None) -> img
# rect: 사각형의 위치 정보 (x, y, w, h)

# cv2.circle(img, center, radius, color, thickness = None, lineType = None) -> img
```

```
# center: 원의 중심좌표 (x, y)
# radius : 원의 반지름

# cv2.polylines(img, pts, isClosed, color, thickness = None, lineType = None)
# center: 다각형 점들의 array
# isClosed : True for 폐곡선
```

In [84]:

```
import numpy as np
import cv2

img = np.full((600, 1200, 3), 255, np.uint8)

# x, y 좌표순서
cv2.line(img, (50, 50), (300, 50), (0, 0, 255), 5)
cv2.line(img, (300, 50), (150, 300), (0, 0, 255), 5)
cv2.line(img, (400, 50), (400, 300), (0, 0, 255), 5)

cv2.rectangle(img, (50, 410, 350, 100), (0, 255, 0), 2)
# cv2.rectangle(img, (70, 220), (180, 280), (0, 128, 0), -1)

cv2.circle(img, (600, 200), 100, (255, 255, 0), 3, cv2.LINE_AA)
# cv2.circle(img, (300, 100), 60, (255, 0, 0), -1, cv2.LINE_8)

# pts = np.array([[250, 200], [300, 200], [350, 300], [250, 300]])
# cv2.polylines(img, [pts], True, (255, 0, 255), 2, cv2.LINE_AA)

text = 'OpenCV, ' + cv2.__version__
cv2.putText(img, text, (800, 100), cv2.FONT_HERSHEY_SIMPLEX, 0.8,
            (0, 0, 255), 1, cv2.LINE_AA)

cv2.imshow("img", img)
cv2.waitKey()
cv2.destroyAllWindows()
```

## 카메라와 동영상 처리하기

In [13]:

```
# cv2.VideoCapture(index/filename, apiPreference=None) -> retval
# index: camera_id or filename
# apiPreference=None

# cv2.VideoCapture.open(index/filename, apiPreference=None) -> retval
# retval: True or False

# cv2.VideoCapture.read(image=None) -> retval, image
# retval : True or False
# image: Frame

# cv2.VideoCapture.get() -> retval

# cv2.Canny(image, threshold1, threshold2)
# image: 입력영상
# threshold1 = low value, threshold2 = high value
```

In [3]:

```
# 카메라 열기
cap = cv2.VideoCapture(0)

if not cap.isOpened(): #True or False
    print("Videocap open failed!")
    cap.release()
    sys.exit()
```

```

# 카메라 프레임 크기 출력

width = round(cap.get(cv2.CAP_PROP_FRAME_WIDTH))
height = round(cap.get(cv2.CAP_PROP_FRAME_HEIGHT))

print('Frame width:', int(cap.get(cv2.CAP_PROP_FRAME_WIDTH)))
print('Frame height:', int(cap.get(cv2.CAP_PROP_FRAME_HEIGHT)))

# cap.set(cv2.CAP_PROP_FRAME_WIDTH, 320)
# cap.set(cv2.CAP_PROP_FRAME_HEIGHT, 240)

# 카메라 프레임 처리
while True:

    ret, frame = cap.read()

    if not ret:
        print('Video read failed')
        break

    #     edge = cv2.Canny(frame, 50, 150)

    #     inversed = cv2.bitwise_not(frame)
    #     inversed = ~frame # 반전

    cv2.imshow('frame', frame)
    #     cv2.imshow('frame1', edge)
    #     cv2.imshow('inversed', inversed)

    if cv2.waitKey(10) == 27:
        break

cap.release()
cv2.destroyAllWindows()

```

Videocap open failed!

An exception has occurred, use %tb to see the full traceback.

### SystemExit

C:\Users\user\Anaconda3\envs\mytf\lib\site-packages\IPython\core\interactiveshell.py:3449: UserWarning: To exit: use 'exit', 'quit', or Ctrl-D.  
warn("To exit: use 'exit', 'quit', or Ctrl-D.", stacklevel=1)

## 카메라 동영상 저장하기

In [15]:

```

# cv2.VideoWriter(filename, fourcc, fps, framesize, isColor=None) -> retval
# filename: 저장할 이름
# fourcc: cv2.VideoWriter_fourcc(*'DIVX') 를 사용
# fps: 초당 프레임 수 e.g. 30
# frameSize: 프레임 크기 e.g., [640, 480])
# isColor: Color 영상
#

```

In [16]:

```

# import sys
# import cv2

```

```

cap = cv2.VideoCapture(0)

if not cap.isOpened():
    print("Camera open failed!")
    sys.exit()

# cap.get(cv2.CAP_PROP_FRAME_WIDTH) -> float type 반환
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(*'MJPG')
fourcc = cv2.VideoWriter_fourcc(*'DIVX') # *'DIVX' == 'D', 'I', 'V', 'X'
# delay = round(1000 / fps)

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

if not out.isOpened():
    print('File open failed!')
    cap.release()
    sys.exit()

while True:
    ret, frame = cap.read()

    if not ret:
        print('Video read failed')
        break

    #     inversed = ~frame
    #     edge = cv2.Canny(frame, 50, 150)
    #     edge_color = cv2.cvtColor(edge, cv2.COLOR_GRAY2BGR)

    out.write(frame) #소리는 capture가 안됨
    #     out.write(inversed)
    #     out.write(edge_color)

    cv2.imshow('frame', frame)
    #     cv2.imshow('inversed', inversed)
    #     cv2.imshow('edge', edge)

    if cv2.waitKey(delay) == 27:
        break

cap.release()
out.release()

cv2.destroyAllWindows()

```

## 키보드 이벤트 처리하기

In [17]:

```

# import sys
# import numpy as np
# import cv2

img = cv2.imread('cat.bmp', cv2.IMREAD_GRAYSCALE)

if img is None:
    print('Image load failed!')
    sys.exit()

```

```

cv2.namedWindow('image')
cv2.imshow('image', img)

# cv2.waitKeyEx()

while True:

    keycode = cv2.waitKey()

    if keycode == ord('i'):
        img = cv2.bitwise_not(img)
        cv2.imshow('image', img)

    elif keycode == 27:
        break

cv2.destroyAllWindows()

```

Image load failed!

An exception has occurred, use %tb to see the full traceback.

### SystemExit

C:\Users\nbumk\anaconda3\envs\mytf\lib\site-packages\IPython\core\interactiveshell.py:3445: UserWarning: To exit: use 'exit', 'quit', or Ctrl-D.  
warn("To exit: use 'exit', 'quit', or Ctrl-D.", stacklevel=1)

## 마우스 이벤트 처리하기

```

In [ ]: # cv2.setMouseCallback(windowName, onMouse, param = None) -> None
        # windowName: 마우스이벤트를 수행할 창 이름
        # onMouse: 마우스 이벤트 콜백함수
        # param: 콜백함수에 전달할 데이터

        # onMouse(event, x, y, flags, param) -> None
        # event: 마우스 이벤트 종류 e.g., cv2.EVENT_LBUTTONDOWN
        # x, y : 창을 기준으로 이벤트 발생좌표
        # flags: 이벤트시 발생 상태 e.g., "ctrl"
        # param: cv2.setMouseCallback() 함수에서 설정한 데이터

```

```

In [ ]: # import sys
        # import numpy as np
        # import cv2

        oldx = oldy = -1

        def on_mouse(event, x, y, flags, param):
            global oldx, oldy

            if event == cv2.EVENT_LBUTTONDOWN:
                oldx, oldy = x, y
                print('EVENT_LBUTTONDOWN: %d, %d' % (x, y))

            elif event == cv2.EVENT_LBUTTONUP:
                print('EVENT_LBUTTONUP: %d, %d' % (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('image', img)

```



```
#                                oldx, oldy = x, y

img = np.ones((480, 640, 3), dtype=np.uint8) * 255

cv2.namedWindow('image')
cv2.setMouseCallback('image', on_mouse, img) #항상 창을 띄우고 호출

cv2.imshow('image', img)
cv2.waitKey()

cv2.destroyAllWindows()
```

## 트랙바 사용하기

```
In [ ]: # createTrackbar(trackbarName, windowName, value, count, onChange) -> None
# trackbarName: 트랙바 이름
# windowName : 트랙바를 생성할 창 이름
# value : 트랙바 위치 초기값
# count : 트랙바 최댓값, 최솟값은 0
# onChange :callback 함수 e.g., onChange(pos) 위치를 정수형태로 전달
```

```
In [ ]: # import numpy as np
# import cv2

def on_level_change(pos):
    print(pos)
    value = pos * 16
    #     if value >= 255:
    #         value = 255
    #     value = np.clip(value,0,255)

    img[:] = value
    cv2.imshow('image', img)

img = np.zeros((480, 640), np.uint8)
cv2.namedWindow('image')
cv2.createTrackbar('level', 'image', 0, 16, on_level_change) # 창이 생성된 후 호출

cv2.imshow('image', img)
cv2.waitKey()
cv2.destroyAllWindows()
```