



1주차 chap02

Chapter2. OpenCV로 시작하는 컴퓨터 비전

2.1 OpenCV 소개

OpenCV : 인텔사에서 만든 컴퓨터 비전 라이브러리, C와 C++로 개발됨

- 교차플랫폼을 사용하기 때문에 개발한 프로그램을 어떤 운영체 플랫폼에서도 사용 가능

2.3 객체지향 잘 활용하기

객체지향의 특성

- 객체 능동적 : .을 찍어 자신이 가진 함수(method, 멤버변수 포함) 능동적 호출
- 필요한 만큼 찍어내기 : 클래스로 객체 무한정 생성

```
import numpy as np

a=np.array([4,5,0,1,2,3,6,7,8,9,10,11])

print(a)
print(type(a)) # 객체가 어떤 클래스인지 확인
print(dir(a)) # 사용가능한 멤버 함수 목록

# 멤버함수 적용
print(a.shape)
a.sort()
print(a)
```

help : 멤버 함수가 무슨 일을 하며 어떻게 사용되는지 알기 위함

ex) help(a.sort)

Help on built-in function sort:

sort(...) method of numpy.ndarray instance
a.sort(axis=-1, kind=None, order=None)

Sort an array in-place. Refer to 'numpy.sort' for full documentation.

Parameters

axis : int, optional

Axis along which to sort. Default is -1, which means sort along the last axis.

kind : {'quicksort', 'mergesort', 'heapsort', 'stable'}, optional

Sorting algorithm. The default is 'quicksort'. Note that both 'stable' and 'mergesort' use timsort under the covers and, in general, the actual implementation will vary with datatype. The 'mergesort' option is retained for backwards compatibility.

.. versionchanged:: 1.15.0

The 'stable' option was added.

order : str or list of str, optional

When 'a' is an array with fields defined, this argument specifies which fields to compare first, second, etc. A single field can be specified as a string, and not all fields need be specified, but unspecified fields will still be used, in the order in which they come up in the dtype, to break ties.

See Also

numpy.sort : Return a sorted copy of an array.

numpy.argsort : Indirect sort.

numpy.lexsort : Indirect stable sort on multiple keys.

numpy.searchsorted : Find elements in sorted array.

numpy.partition: Partial sort.

Notes

See 'numpy.sort' for notes on the different sorting algorithms.

Examples

```
>>> a = np.array([[1,4], [3,1]])
```

```
>>> a.sort(axis=1)
```

```
>>> a
```

```
array([[1, 4],  
       [1, 3]])
```

```
>>> a.sort(axis=0)
```

```
>>> a
```

```
array([[1, 3],  
       [1, 4]])
```

Use the 'order' keyword to specify a field to use when sorting a structured array:

```
>>> a = np.array([('a', 2), ('c', 1)], dtype=[('x', 'S1'), ('y', int)])
```

```
>>> a.sort(order='y')
```

```
>>> a
```

2.4 영상을 읽고 표시하기

```
import cv2 as cv
import sys

img = cv.imread("glory.jpg") # 영상읽기

if img is None:
    sys.exit("파일을 찾을 수 없습니다.")

cv.imshow("Image Display", img) # 윈도우에 영상표시

cv.waitKey() # 키 입력대기시간 ms
cv.destroyAllWindows() # 윈도우 창 종료 함수
```

- if 구문 역할 : 파일이 없거나 읽는 도중 오류 발생시 img None
- imshow : 윈도우에 영상표시 역할 (윈도우 이름, 디스플레이할 영상)
- waitKey : 키가 눌리면 해당 키의 유니코드 값 반환, 기본값으로 없거나 0
 - 지정시간 내에 키 입력이 일어나지 않으면 -1 반환하며 아래로 내려가 윈도우 종료
 - destroyAllWindows : 키 입력 대기하다가 윈도우 닫고 정상적으로 프로그램 종료

오류 해결 방법

- colab :

cv2.imshow() is disabled in Colab, because it causes Jupyter sessions

코랩에서 위 옵션이 비활성화

google colab 자체 개발 라이브러리 적용

```
from google.colab.patches import cv2_imshow
```
- jupyter : !pip install opencv-python 로 오류해결

np 클래스형 객체

- np는 다차원 배열을 위한 표준 모듈이기에 사용
- 3차원 배열, 색 표현 **BGR**
 - 색상을 표현하는 방법으로는 RGB(Red, Green, Blue) 방식. 빨강, 초록, 파랑 세 가지 색의 빛을 섞어서 원하는 색 형성.

- 각 색상은 0~255 사이의 값으로 표시하고 값이 커질수록 해당 색상의 빛이 밝아지는 원리
- RGB = (255, 255, 255) 일 때는 흰색이고, RGB = (0, 0, 0) 일 때는 검정.
- 그러나 **OpenCV는 그 반대의 순서인 BGR로 표현합니다**

OpenCV - 7. 이미지 색상 표현 방식(BGR, HSV, YUV)_(tistory.com)

```
print(img[0,0,0], img[0,0,1], img[0,0,2]) # (0,0) 화소 조사
print(img[0,1,0], img[0,1,1], img[0,1,2]) # (0,1) 화소 조사
```

화소(픽셀)은 좌표로 표기하며 영상을 구성하는 한 점

좌표(BGR)이 색깔을 의미

2.5 영상 형태 변환 & 크기 축소

```
import cv2 as cv
import sys

img = cv.imread("glory.jpg") # 영상읽기

if img is None:
    sys.exit("파일을 찾을 수 없습니다.")

gray = cv.cvtColor(img,cv.COLOR_BGR2GRAY) # BGR 컬러 영상을 명암 영상으로 변환
gray_small=cv.resize(gray,dsiz=(0,0),fx=0.5,fy=0.5) # 반으로 축소

cv.imwrite("glory_gray.jpg",gray) # 영상을 파일에 저장
cv.imwrite("glory_gray_small.jpg",gray_small)

# 윈도우에 영상표시 : 영상 3개를 서로 다른 윈도우 3개에 디스플레이

cv.imshow('Color image',img)
cv.imshow('Gray image',gray)
cv.imshow('Gray image small',gray_small)

cv.waitKey()
cv.destroyAllWindows()
```

- cvtColor : 컬러영상을 명암영상으로 변환

RGB ↔ GRAY

Transformations within RGB space like adding/removing the alpha channel, reversing the channel order, conversion to/from 16-bit RGB color (R5:G6:B5 or R5:G5:B5), as well as conversion to/from grayscale using:

$$\text{RGB[A] to Gray: } Y \leftarrow 0.299 \cdot R + 0.587 \cdot G + 0.114 \cdot B$$

and

$$\text{Gray to RGB[A]: } R \leftarrow Y, G \leftarrow Y, B \leftarrow Y, A \leftarrow \max(\text{ChannelRange})$$

The conversion from a RGB image to gray is done with:

Opencv 컬러 이미지 Gray 변환(cvtColor 함수) : 네이버 블로그 (naver.com)

- 컬러기반 RGB를 명암영상으로 변환후, round 함수를 이용해 반올림
 - 이러한 연산을 모든 화소에 적용하면 명암 영상 생성
-
- resize : 크기변환 (입력영상, 변환할 크기지정, 가로, 세로 비율)

+) 영상의 확장자를 변경하고 싶다면, imwrite에서 변경해 파일 저장

ex. glory_gray.jpg → glory_gray.png

2.6 웹 캠에서 비디오 읽기

```
import cv2 as cv
import sys

cap = cv.VideoCapture(0, cv.CAP_DSHOW) # 카메라와 연결시도

if not cap.isOpened():
    sys.exit("카메라 연결 실패")

while True:
    ret, frame = cap.read() # 비디오 구성하는 프레임 획득

    if not ret:
        print("프레임 획득에 실패하여 루프를 나갑니다.")
        break

    cv.imshow("Video display", frame)

    key = cv.waitKey(1) # 1밀리초 동안 키보드 입력 기다림
    if key == ord('q'): # 'q'키가 돌아오면 루프 아웃
        break

cap.release() # 카메라와 연결 끊음
cv.destroyAllWindows() # 윈도우 모두 닫음
```

+) 웹캠 사용이 안되는 이유 :

윈도우 기본 카메라 앱에서는 출력되지만, opencv에서 특정 웹캠 연결 문제가 발생하여 영상이 출력되지 않는 경우가 가끔 발생합니다. 다양한 이유가 있지만, 동영상 프레임을 불러오는 과정을 결정짓는 **apiPreference 문제** 로 발생하기도 합니다.

[Python] OpenCV 웹캠 연결 문제 cv2.VideoCapture 해결 방법 #1 (tistory.com)

- VideoCapture : 웹캠과 연결 시도 (웹 캠 번호, 비디오 화면 보임)
- 웹캠 연결 여부 확인 : 실패시 isOpened 함수가 fail값을 보유하게 됨
- 동영상 입력하는 방법
 - 사용자가 q 누르면 루프 out
 - read : 호출한 순간의 영상 한 장, 프레임 획득 및 성공 여부와 함께 프레임 반환
 - ret : 성공여부
 - frame : 프레임 저장
 - Video display : 윈도우에 획득한 영상 디스플레이
- waitKey : 인수를 크게 설정해 기다리는 시간이 길어지면 지연 발생
→ 비디오 매끄럽지 않게 나타나는 부작용 발생

비디오에서 수집한 영상 이어붙이기

```
import cv2 as cv
import numpy as np
import sys

cap=cv.VideoCapture(0,cv.CAP_DSHOW) # 카메라와 연결 시도

if not cap.isOpened():
    sys.exit('카메라 연결 실패')

frames=[]
while True:
    ret,frame = cap.read() # 비디오를 구성하는 프레임 획득
```

```

if not ret:
    print('프레임 획득에 실패하여 루프를 나갑니다.')
    break

cv.imshow("Video display", frame)

key = cv.waitKey(1) # 1밀리초 동안 키보드 입력 기다림
if key==ord('c'): # 'c' 키가 들어오면 프레임을 리스트에 추가
    frames.append(frame) # 'q' 키가 들어오면 루프를 빠져나감
elif key==ord('q'):
    break

cap.release() # 카메라와 연결을 끊음
cv.destroyAllWindows()

if len(frames)>0: # 수집된 영상이 있으면
    imgs=frames[0]
    for i in range(1,min(3, len(frames))): # 최대 3개까지 이어 붙임.
        imgs=np.hstack((imgs, frames[i]))

cv.imshow("collected images",imgs)

cv.waitKey()
cv.destroyAllWindows()

```

frames=[]

수집한 영상을 저장할 리스트 생성

```

if key==ord('c'): # 'C' 키가 들어오면 프레임을 리스트에 추가
    frames.append(frame)

```

사용자가 c를 누를 때마다 획득한 프레임을 리스트 추가

```

if len(frames)>0: # 수집된 영상이 있으면
    imgs=frames[0]
    for i in range(1,min(3, len(frames))): # 최대 3개까지 이어 붙임.
        imgs=np.hstack((imgs, frames[i]))

cv.imshow("collected images",imgs)

cv.waitKey()
cv.destroyAllWindows()

```

- hstack : 배열을 이어 붙이는 함수

수집한 영상을 이어 붙여 윈도우에 표시하며 최대 세개 붙이기

한정적인 이유 : 영상이 너무 커서 화면 밖으로 나가는 것 방지를 위함

ex 8번 눌러도 세번 반복해 배열 3개 붙임

frame리스트와 imgs 배열로 영상의 특성 확인

→ type과 shape 잘 이해하기

2.7 그래픽 기능과 사용자 인터페이스 만들기

line 직선그리기

rectangle 직사각형 그리기

polylines 다각형 그리기

circle 원 그리기

ellipse 타원 그리기

putText 문자열 작성

직사각형 그리기

```
cv.rectangle(img, (830, 30), (1000, 200), (0, 0, 255), 2)
```

직사각형 그릴 영상, 직사각형의 왼쪽 위 구석점 좌표, 오른쪽 아래 구석점 좌표, 색지정 (BGR), 선두께

문자열 작성

```
cv.putText(img, 'laugh', (830, 24), cv.FONT_HERSHEY_SIMPLEX, 1, (255, 0, 0), 2)
```

영상, 문자열, 문자열의 왼쪽 아래 구석점 위치 지정, 폰트 종류, 글자 크기, 색, 글자 두께

마우스를 통한 상호작용

```
import cv2 as cv
import sys
```



```

img = cv.imread("glory.jpg") # 영상읽기

if img is None:
    sys.exit("파일을 찾을 수 없습니다.")

def draw(event,x,y,flags,param): # 콜백 함수
    if event==cv.EVENT_LBUTTONDOWN:# 마우스 왼쪽 버튼 클릭했을 때
        cv.rectangle(img,(x,y),(x+200,y+200),(0,0,255),2)
    elif event==cv.EVENT_RBUTTONDOWN: # 마우스 오른쪽 버튼 클릭
        cv.rectangle(img,(x,y),(x+100,y+100),(255,0,0),2)

    cv.imshow("Drawing",img)

cv.namedWindow("Drawing")
cv.imshow("Drawing",img)

cv.setMouseCallback('Drawing',draw) # drawing 윈도우에 draw 콜백 함수 지정

while(True): # 마우스 이벤트가 언제 발생할지 모르기 때문에 무한 반복
    if cv.waitKey(1)==ord('q'):
        cv.destroyAllWindows()
        break

```