
파이썬

OpenCV 프로그래밍 2

목차

- OpenCV 프로그래밍 기초
 - ✓ 영상 불러와서 출력하기
 - ✓ 영상 데이터 다루기 (생성, 복사, 크롭, 합성)
 - ✓ 컬러 영상 처리: 색 공간 변환
 - ✓ 기하학적 변환 (크기 변환, 회전 변환)
 - ✓ OpenCV 그리기 함수

- 카메라 & 동영상 처리
 - ✓ 카메라 & 동영상 프레임 처리
 - ✓ 동영상 파일 저장하기

영상 불러와서 출력하기

□ BMP 파일을 불러와서 출력하는 소스 코드 추가 입력

```
import sys
import cv2
```

```
img = cv2.imread('cat.bmp')
```

cat.bmp 파일을 불러와 img 변수에 저장.
type(img): <class 'numpy.ndarray'>

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

영상 파일 불러오기가 실패하면 img가 None 이고, 이 경우 메시지를 출력하고 종료.

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

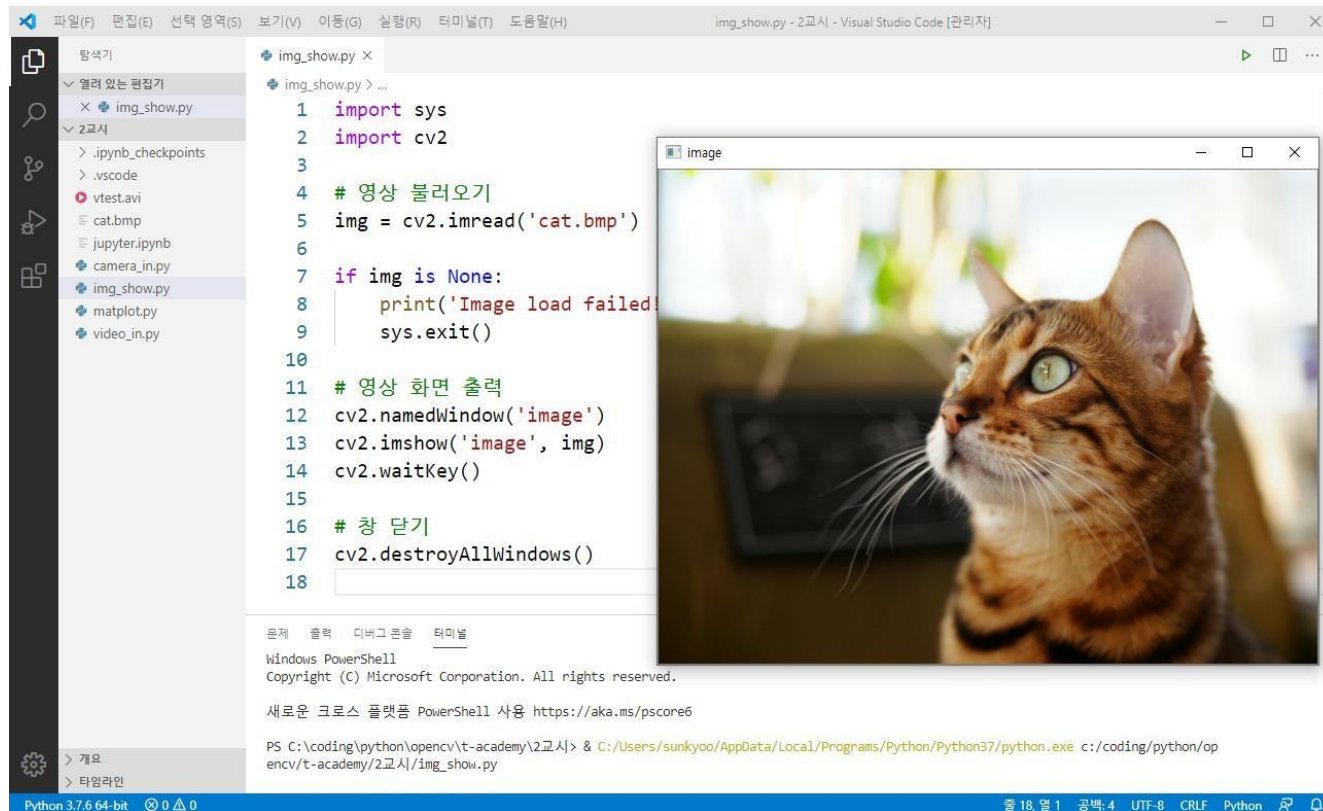
image라는 이름의 새 창을 만들고,
이 창에 img 영상을 출력.
그리고 키보드 입력이 있을 때까지 대기.

```
cv2.destroyAllWindows()
```

생성된 모든 창을 닫음

영상 불러와서 출력하기

- img_show.py 프로그램 실행하기
 - ✓ cat.bmp 파일을 해당 폴더로 복사한 후, 프로그램 실행



OpenCV API

□ 영상 파일 불러오기

```
cv2.imread(filename, flags=None) -> retval
```

- ✓ filename: 불러올 영상 파일 이름 (문자열)
 - 상대 경로: 'cat.bmp', '../data/cat.bmp'
 - 절대 경로: 'c:\cat.bmp', '/home/id/cat.bmp'
- ✓ flags: 영상 파일 불러오기 옵션 플래그

cv2.IMREAD_COLOR	BGR 컬러 영상으로 읽기 (기본값) shape = (rows, cols, 3), dtype=uint8
cv2.IMREAD_GRAYSCALE	그레이스케일 영상으로 읽기 shape = (rows, cols), dtype=uint8
cv2.IMREAD_UNCHANGED	영상 파일 속성 그대로 읽기 (e.g.) 투명한 PNG 파일: shape = (rows, cols, 4)

- ✓ retval: 불러온 영상 데이터 ([numpy.ndarray](#))

OpenCV API

□ 영상 파일 저장하기

```
cv2.imwrite(filename, img, params=None) -> retval
```

- ✓ filename: 저장할 영상 파일 이름 (문자열)
- ✓ img: 저장할 영상 데이터 (numpy.ndarray)
- ✓ params: 파일 저장 옵션 지정 (속성 & 값의 정수 쌍)
 - e.g) JPG 파일 압축률을 90%로 지정하고 싶다면
[cv2.IMWRITE_JPEG_QUALITY, 90] 지정
- ✓ retval: 정상적으로 저장하면 True, 실패하면 False.

OpenCV API

□ 새 창 띄우기 & 창 닫기

```
cv2.namedWindow(winname, flags=None) -> None
```

- ✓ winname: 창 고유 이름. 이 이름으로 창을 구분함.
- ✓ flags: 창 속성 지정 플래그

cv2.WINDOW_NORMAL	영상 크기가 창 크기에 맞게 지정됨
cv2.WINDOW_AUTOSIZE	창 크기가 영상 크기에 맞게 자동으로 변경됨 (기본값)

```
cv2.destroyWindow(winname) -> None
```

```
cv2.destroyAllWindows() -> None
```

- ✓ winname: 닫고자 하는 창 이름
- ✓ 일반적인 경우 프로그램 종료 시 운영 체제에 의해 열려 있는 모든 창이 자동으로 닫힘

OpenCV API

□ 창 위치 & 크기 지정

```
cv2.moveWindow(winname, x, y) -> None
```

- ✓ winname: 창 이름
- ✓ x,y: 이동할 위치 좌표

```
cv2.resizeWindow(winname, width, height) -> None
```

- ✓ winname: 창 이름
- ✓ width,height: 변경할 창 크기
- ✓ 참고 사항
 - 창 생성 시 cv2.WINDOW_NORMAL 속성으로 생성되어야 동작함.
 - 영상 출력 부분의 크기만을 고려함 (제목 표시줄, 창 경계는 고려되지 않음)

OpenCV API

□ 영상 출력하기

```
cv2.imshow(winname, mat) -> None
```

- ✓ winname: 영상을 출력할 대상 창 이름
- ✓ mat: 출력할 영상 데이터 ([numpy.ndarray](#))
- ✓ 데이터 타입에 따른 출력 방식
 - uint8: 픽셀 값을 그대로 출력
 - uint16, int16: 픽셀 값을 255로 나눠서 출력
 - float32, float64: 픽셀 값에 255를 곱해서 출력
- ✓ 참고 사항
 - 만약 winname에 해당하는 창이 없으면 자동으로 cv2.WINDOW_AUTOSIZE 속성의 창을 새로 만들어서 영상을 출력함
 - Windows 운영체제에서는 Ctrl + C (복사), Ctrl + S (저장) 지원
 - 실제로는 cv2.waitKey() 함수를 호출해야 화면에 영상이 나타남.

OpenCV API

□ 키보드 입력 대기

```
cv2.waitKey(delay=None) -> retval
```

- ✓ delay: 밀리초 단위 대기 시간. $\text{delay} \leq 0$ 이면 무한히 기다림.
기본값은 0.
- ✓ retval: 눌린 키 값(ASCII code). 키가 눌리지 않으면 -1.
- ✓ 참고 사항
 - cv2.waitKey() 함수는 OpenCV 창이 하나라도 있을 때 동작함
 - 특정 키 입력을 확인하려면 ord() 함수를 이용

```
while True:  
    if cv2.waitKey() == ord('q'):  
        break
```

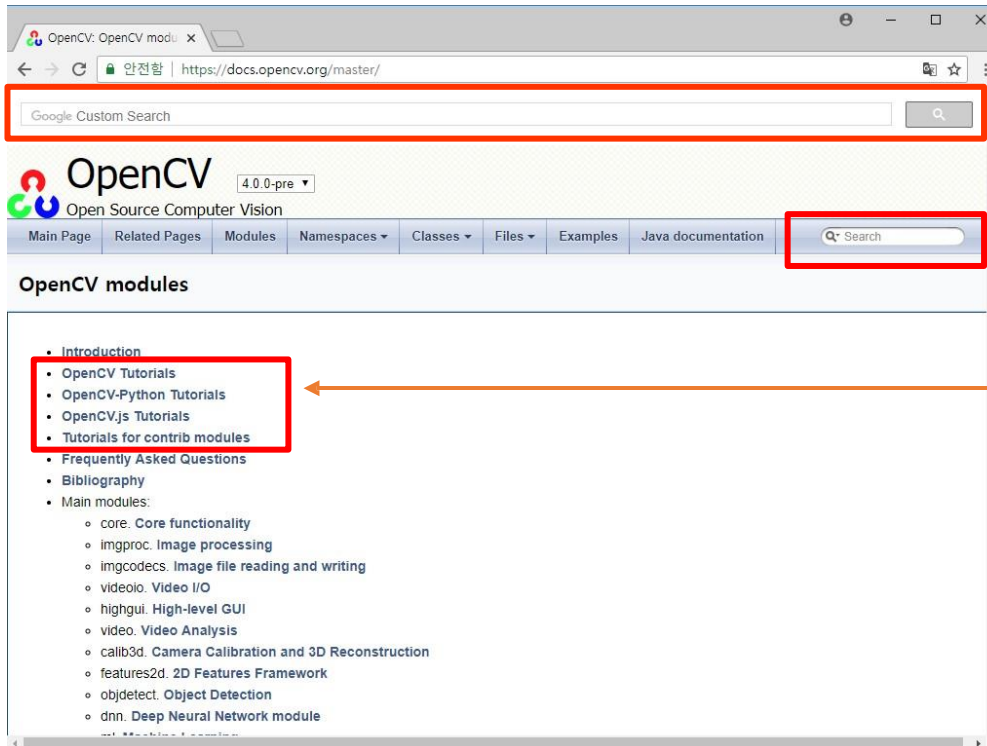
- 주요 특수키 코드: ESC ⑦ 27, ENTER ⑦ 13, TAB ⑦

9

OpenCV API 도움말 찾기

□ OpenCV API 도움말 찾기

- ✓ OpenCV 최신 도움말: <http://docs.opencv.org/master/>
- ✓ OpenCV 도움말 웹페이지에서 우측 상단 검색창 활용



OpenCV 사이트
내부에서 구글 검색

OpenCV API 검색

Tutorials

Matplotlib을 이용한 영상 출력

□ Matplotlib 패키지

- ✓ 함수 그래프, 차트(chart), 히스토그램(histogram) 등의 다양한 그리기 기능을 제공하는 Python 패키지
- ✓ 보통 `matplotlib.pyplot` 모듈을 `plt` 이름으로 불러와서 사용
- ✓ Jupyter Notebook에서 사용할 때 유용

```
> pip install matplotlib
```

□ 컬러 영상 출력

- ✓ 컬러 영상의 색상 정보가 **RGB 순서**이어야 함
- ✓ `cv2.imread()` 함수로 불러온 영상의 색상 정보는 **BGR 순서**이므로 이를 RGB 순서로 변경해야 함 ⑦ `cv2.cvtColor()` 함수 이용

□ 그레이스케일 영상 출력

- ✓ `plt.imshow()` 함수에서 컬러맵을 `cmap='gray'` 으로 지정

Matplotlib을 이용한 영상 출력

```
import matplotlib.pyplot as plt
import cv2

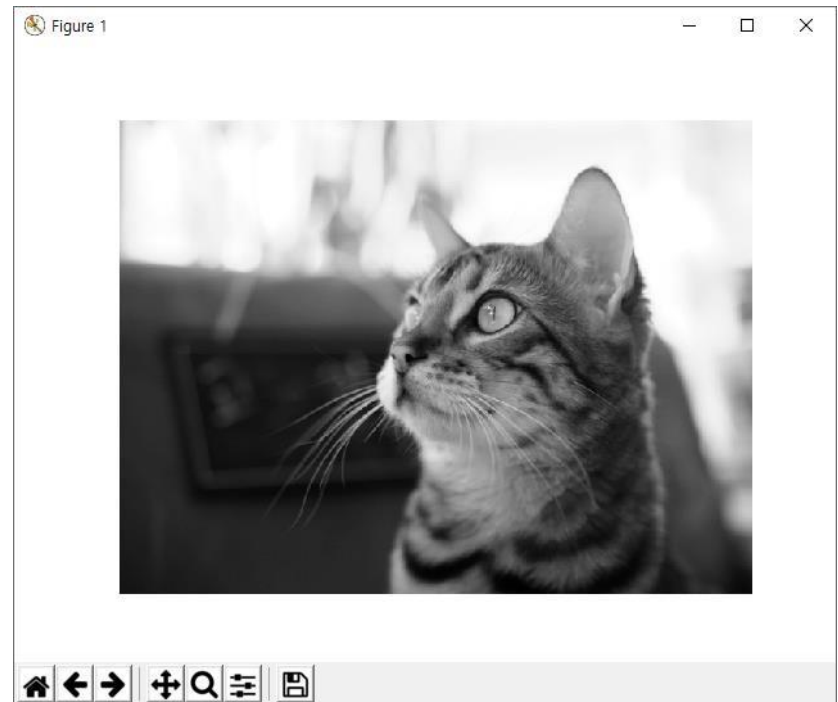
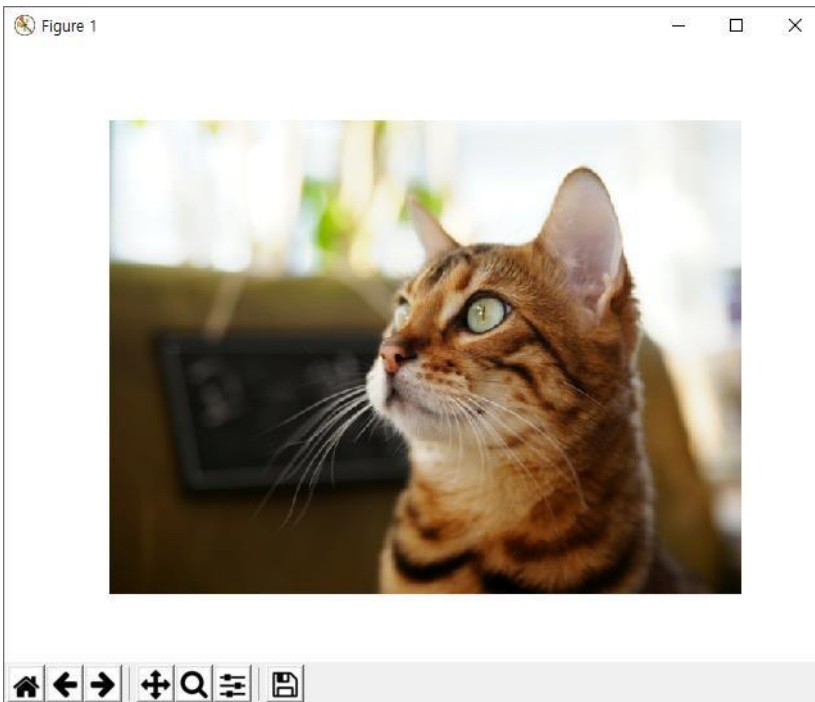
# 컬러 영상 출력
imgBGR = cv2.imread('cat.bmp')
imgRGB = cv2.cvtColor(imgBGR, cv2.COLOR_BGR2RGB)
plt.axis('off')
plt.imshow(imgRGB)
plt.show()

# 그레이스케일 영상 출력
imgGray = cv2.imread('cat.bmp', cv2.IMREAD_GRAYSCALE)
plt.axis('off')
plt.imshow(imgGray, cmap='gray')
plt.show()

# 두 개의 영상을 함께 출력
plt.subplot(121), plt.axis('off'), plt.imshow(imgRGB)
plt.subplot(122), plt.axis('off'), plt.imshow(imgGray, cmap='gray')
plt.show()
```

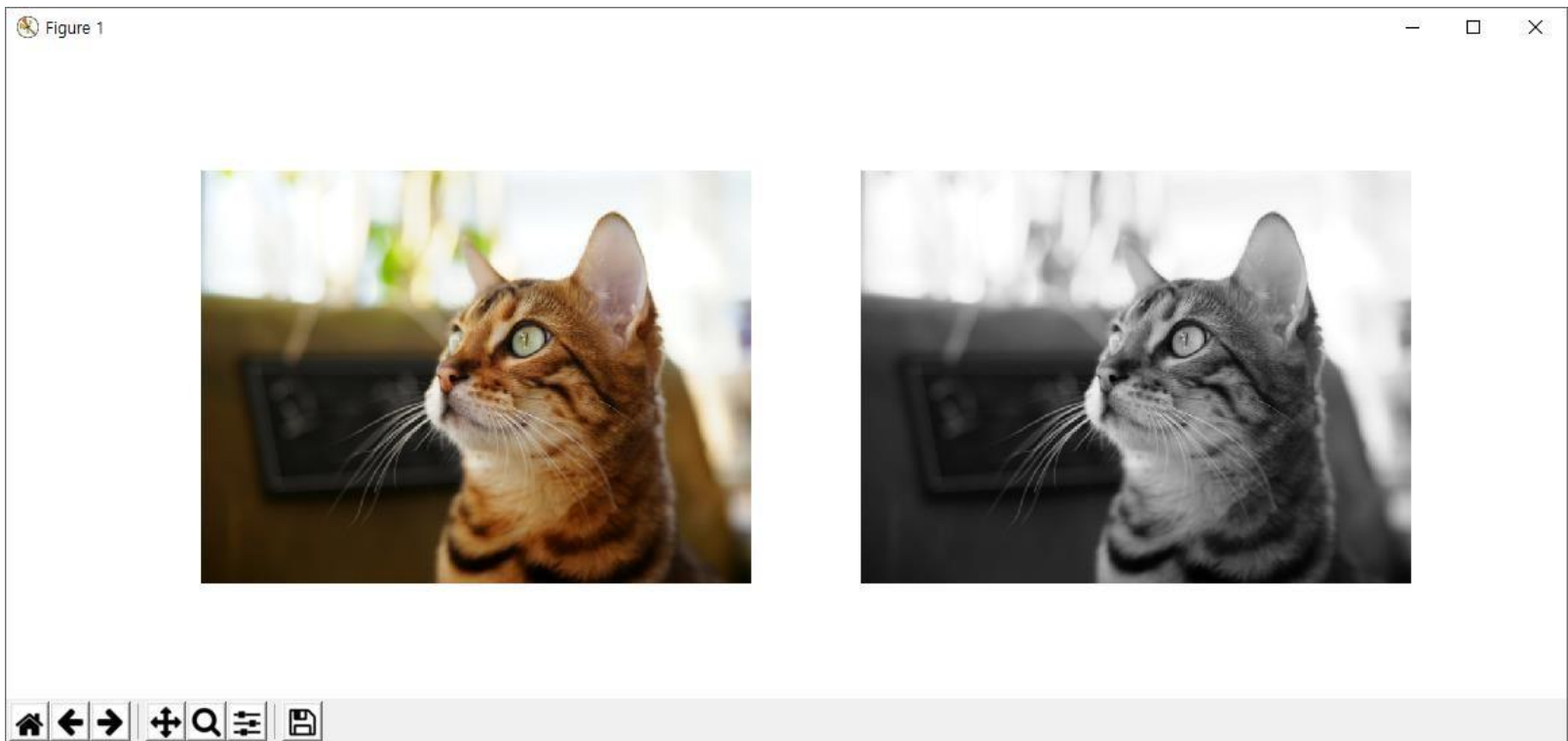
Matplotlib을 이용한 영상 출력

- Matplotlib.pyplot을 이용한 영상 출력 결과



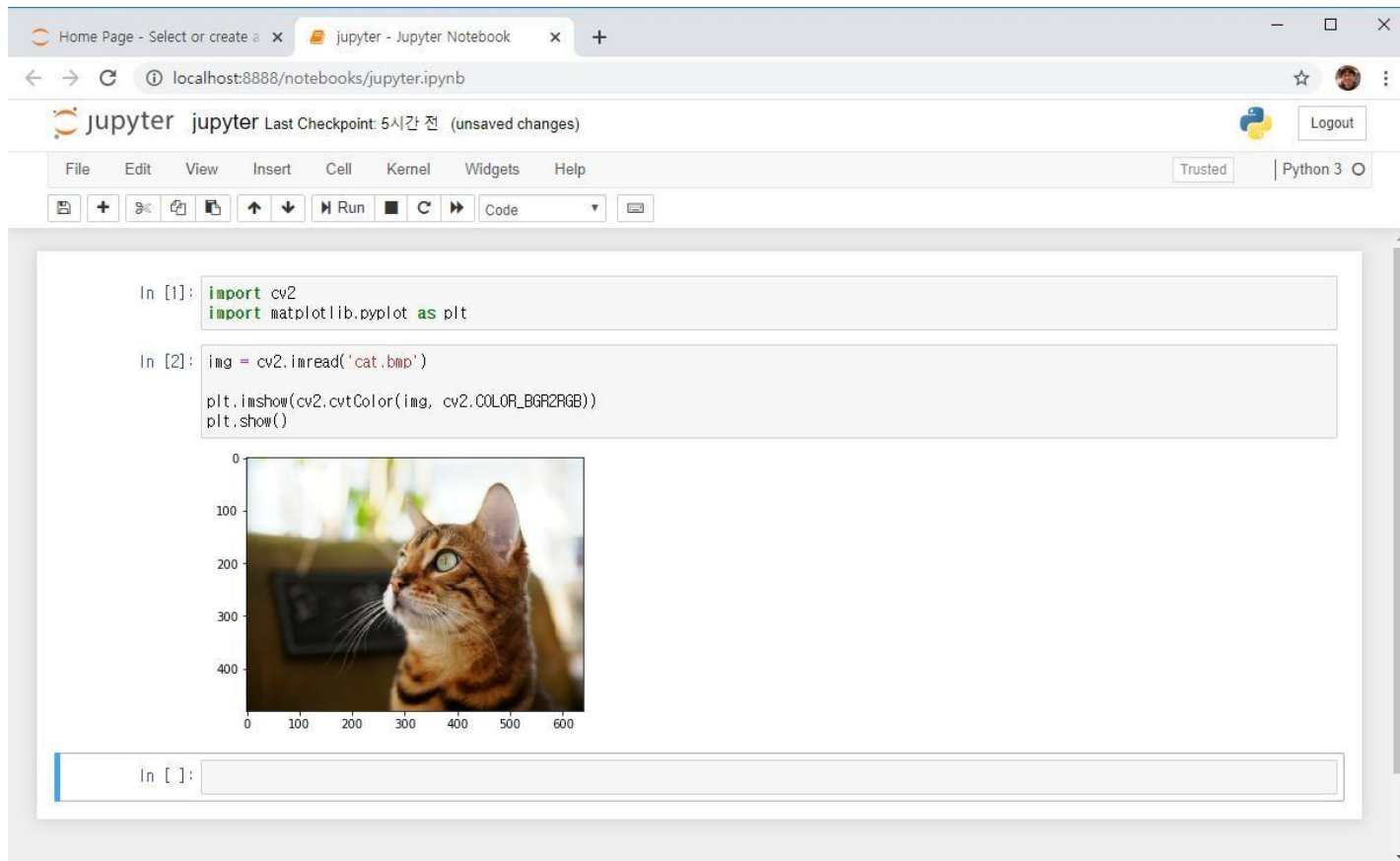
Matplotlib을 이용한 영상 출력

- Matplotlib.pyplot을 이용한 영상 출력 결과



Matplotlib을 이용한 영상 출력

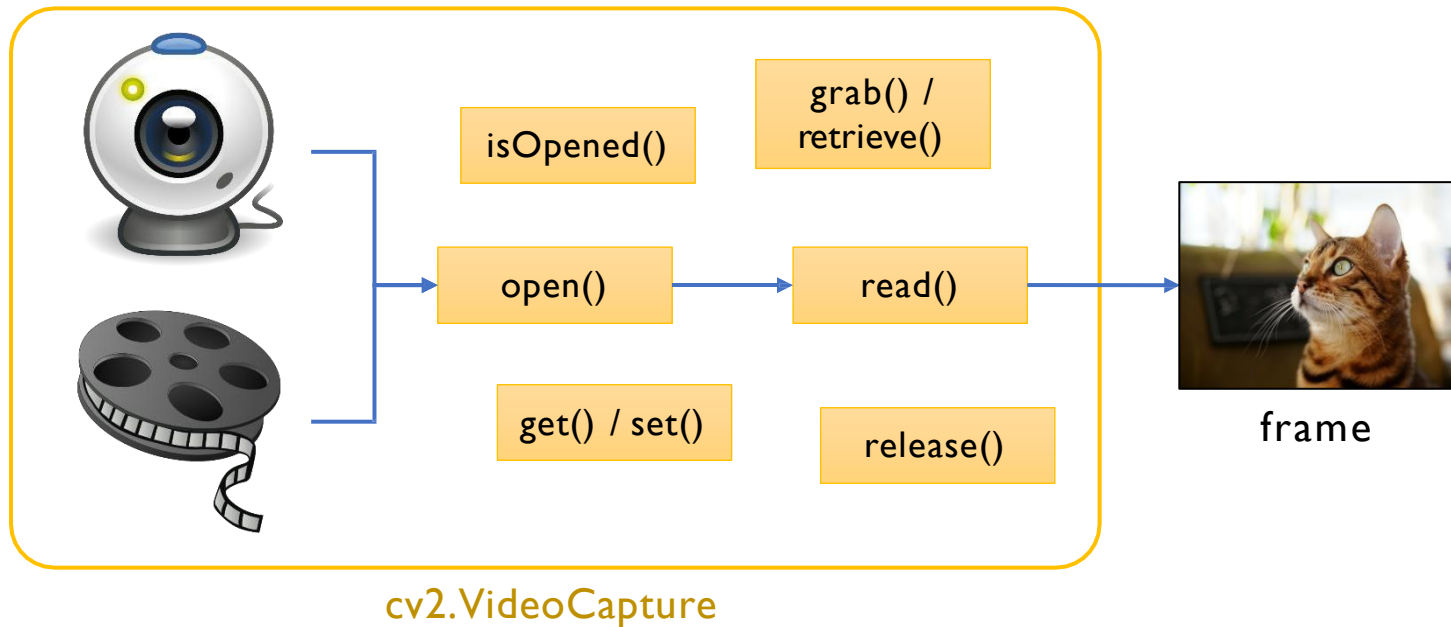
□ Jupyter Notebook에서 영상 출력



카메라 & 동영상 처리

□ cv2.VideoCapture 클래스

- ✓ OpenCV에서는 카메라와 동영상으로부터 프레임(frame)을 받아오는 작업을 cv2.VideoCapture 클래스 하나로 처리함



OpenCV API

□ 동영상, 정지 영상 시퀀스, 비디오 스트림 열기

```
cv2.VideoCapture(filename, apiPreference=None) -> retval
```

- ✓ filename:
 - 비디오 파일 이름 (e.g. 'video.avi')
 - 정지 영상 시퀀스 (e.g. 'img_%02d.jpg')
 - 비디오 스트림 URL (e.g. 'protocol://host:port/script?params|auth')
- ✓ apiPreference: 선호하는 동영상 처리 방법을 지정
- ✓ retval: cv2.VideoCapture 객체

```
cv2.VideoCapture.open(filename, apiPreference=None) -> retval
```

- ✓ retval: 성공하면 True, 실패하면 False.

OpenCV API

□ 카메라 열기

```
cv2.VideoCapture(index, apiPreference=None) -> retval
```

- ✓ index: camera_id + domain_offset(CAP_*) id
 - camera_id == 0 이면 시스템 기본 카메라
 - domain_offset == 0 이면 auto detect.
 - 기본 카메라를 기본 방법으로 열려면 index에 0을 전달.
- ✓ apiPreference: 선호하는 카메라 처리 방법을 지정
- ✓ retval: cv2.VideoCapture 객체

```
cv2.VideoCapture.open(index, apiPreference=None) -> retval
```

- ✓ retval: 성공하면 True, 실패하면 False.

OpenCV API

□ 비디오 캡처가 준비되었는지 확인

```
cv2.VideoCapture.isOpened() -> retval
```

✓ retval: 성공하면 True, 실패하면 False.

□ 프레임 받아오기

```
cv2.VideoCapture.read(image=None) -> retval, image
```

✓ retval: 성공하면 True, 실패하면 False.

✓ image: 현재 프레임 (numpy.ndarray)

OpenCV API

□ 카메라, 비디오 장치 속성 값 참조

```
cv2.VideoCapture.get(propId) -> retval
```

✓ propId: 속성 상수

cv2.CAP_PROP_FRAME_WIDTH	프레임 가로 크기
cv2.CAP_PROP_FRAME_HEIGHT	프레임 세로 크기
cv2.CAP_PROP_FPS	초당 프레임 수
cv2.CAP_PROP_FRAME_COUNT	비디오 파일의 총 프레임 수
cv2.CAP_PROP_POS_MSEC	밀리초 단위로 현재 위치
cv2.CAP_PROP_POS_FRAMES	현재 프레임 번호
cv2.CAP_PROP_EXPOSURE	노출
cv2.CAP_PROP_ZOOM	줌(확대/축소 비율)

✓ retval: 성공하면 해당 속성 값, 실패하면 0.

OpenCV API

□ 카메라, 비디오 장치 속성 값 참조

```
cv2.VideoCapture.set(propId, value) -> retval
```

- ✓ propId: 속성 상수
- ✓ value: 속성 값
- ✓ retval: 성공하면 True, 실패하면 False.

카메라 처리 예제

```
import cv2
```

```
cap = cv2.VideoCapture()  
cap.open(0)
```

VideoCapture 객체 생성 후 0번 카메라 장치 열기.
cap = cv2.VideoCapture(0) 과 동일

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

```
while True:
```

```
    ret, frame = cap.read()
```

카메라로부터 한 프레임을 정상적으로 받아오면
ret에는 True, frame에는 해당 프레임이 저장됨.

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

```
    cv2.imshow('frame', frame)  
    cv2.imshow('edge', edge)
```

현재 프레임과 에지 검출 영상을 출력

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

일정 시간(e.g. 10ms) 기다린 후 다음 프레임 처리.
만약 ESC 키를 누르면 while 루프 종료.

```
cap.release() cv2.destroyAllWindows()
```

사용한 자원을 해제. (생략 가능)

카메라 처리 예제

```
import cv2
```

```
cap = cv2.VideoCapture()  
cap.open(0)
```

```
if not cap.isOpened(): print("C  
amera open failed!") exit()
```

```
print('Frame width:', round(cap.get(cv2.  
print('Frame height:', round(cap.get(cv2.CAP_PROP_FRAME_HEIGHT))  
)
```

```
while True:
```

```
    ret, frame = cap.read()
```

```
if not ret:  
    break
```

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

```
    cv2.imshow('frame', frame)
```

```
    cv2.imshow('edge', edge)
```

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

```
cap.release() cv2.dest  
royAllWindows()
```


동영상 처리 예제

```
import cv2

cap = cv2.VideoCapture('vtest.avi')

fps = round(cap.get(cv2.CAP_PROP_FPS))
delay = round(1000 / fps)

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

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

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

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

cap.release()
cv2.destroyAllWindows()
```

동영상 처리 예제



동영상 파일 저장하기

□ cv2.VideoWriter 클래스

- ✓ OpenCV에서는 cv2.VideoWriter 클래스를 이용하여 일련의 프레임을 동영상 파일로 저장할 수 있음
- ✓ 일련의 프레임은 모두 크기와 데이터 타입이 같아야 함

□ Fourcc (4-문자 코드, four character code)

- ✓ 동영상 파일의 코덱, 압축 방식, 색상 혹은 픽셀 포맷 등을 정의하는 정수 값
- ✓ <http://www.fourcc.org/codecs.php>

cv2.VideoWriter_fourcc(*'DIVX')	DIVX MPEG-4 코덱
cv2.VideoWriter_fourcc(*'XVID')	XVID MPEG-4 코덱
cv2.VideoWriter_fourcc(*'FMP4')	FFMPEG MPEG-4 코덱
cv2.VideoWriter_fourcc(*'X264')	H.264/AVC 코덱
cv2.VideoWriter_fourcc(*'MJPG')	Motion-JPEG 코덱

OpenCV API

□ 저장을 위한 동영상 파일 열기

```
cv2.VideoWriter(filename, fourcc, fps, frameSize, isColor=None)  
-> retval
```

- ✓ filename: 비디오 파일 이름 (e.g. 'video.mp4')
- ✓ fourcc: fourcc (e.g. cv2.VideoWriter_fourcc(*'DIVX'))
- ✓ fps: 초당 프레임 수 (e.g. 30)
- ✓ frameSize: 프레임 크기 (e.g. [640, 480])
- ✓ isColor: 컬러 영상이면 True, 그렇지 않으면 False.
- ✓ retval: cv2.VideoWriter 객체

```
cv2.VideoWriter.open(filename, fourcc, fps, frameSize,  
                      isColor=None) -> retval
```

- ✓ retval: 성공하면 True, 실패하면 False.

OpenCV API

□ 비디오 파일이 준비되었는지 확인

```
cv2.VideoWriter.isOpened() -> retval
```

✓ retval: 성공하면 True, 실패하면 False.

□ 프레임 받아오기

```
cv2.VideoWriter.write(image) -> None
```

✓ image: 저장할 프레임 (numpy.ndarray)

동영상 파일 저장하기

- 카메라 입력 프레임에서 에지 영상을 구하여 동영상으로 저장하기

```
import sys
import cv2

# 카메라로부터 cv2.VideoCapture 객체 생성
cap = cv2.VideoCapture(0)

if not cap.isOpened(): print("C
    amera open failed!") sys.ex
    it()

# 동영상 저장을 위한 cv2.VideoWriter 객체 생성
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') # *'DIVX' == 'D','I','V','X'
out = cv2.VideoWriter('output.avi', fourcc, fps, (w, h))
```

동영상 파일 저장하기

□ 카메라 입력 프레임에서 에지 영상을 구하여 동영상으로 저장하기 (Con't)

```
# 매 프레임 처리 및 화면 출력
while True:
    ret, frame = cap.read()
    if not ret:
        break

    edge = cv2.Canny(frame, 50, 150)
    edge = cv2.cvtColor(edge, cv2.COLOR_GRAY2BGR)
    out.write(edge)

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

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

# 자원 해제 cap
.release() o
ut.release()
cv2.destroyAllWindows()
```