

OpenCV 인터페이스 기초

한림대학교 소프트웨어융합대학
박섭형

2022년 1학기

1 OpenCV 인터페이스 기초

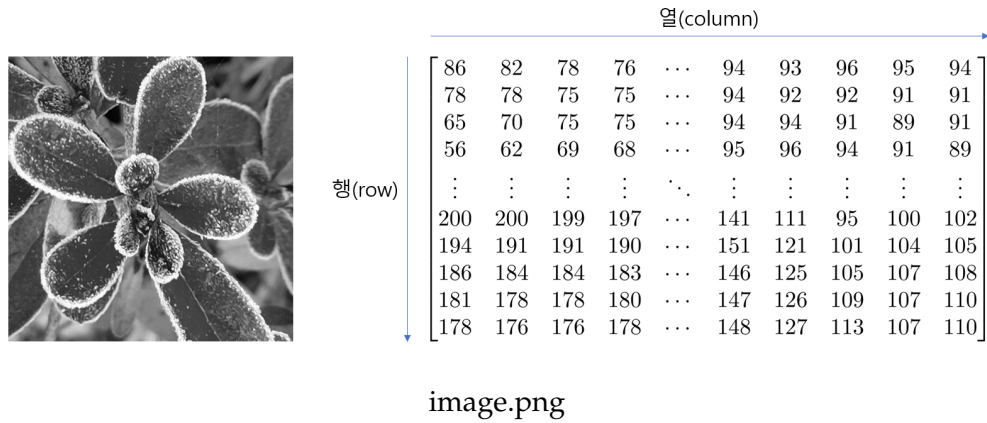
1.1 배울 내용

- 윈도우 제어 함수
- 이벤트 처리 함수
- 그리기 함수
- 영상파일 처리
- 비디오 처리
- Matplotlib 패키지 활용

1.2 사용자 인터페이스 (UI: User Interface)

- 영상 처리 또는 컴퓨터 비전 애플리케이션에서 결과 영상이나 데이터를 효과적으로 보여주는 것이 중요
- 사용자와 컴퓨터 애플리케이션이 상호작용하는 지점
- 효율적인 UI의 목표
 - 사용자의 경험을 쉽고 직관적이 되도록 한다
 - 사용자가 원하는 최대 성능을 얻을 수 있게 하되 사용자의 노력을 최소화한다
- GUI(graphical user interface)
 - 촉각 UI 입력 (키보드나 마우스)
 - 시각적 UI 출력 (모니터)

1.3 영상의 행렬 표현



```
[4]: import cv2
import numpy as np
cv2.__version__
```

[4]: '4.5.5'

1.4 윈도우 제어

- 영상처리
 - 2차원 행렬에 대한 연산
 - 연산과정에서 행렬 원소 변경
 - 전체 영상에 대한 변화 인지하기 어려움
- 윈도우 영상 표시
 - 영상처리로 적용된 행렬 연산의 의미 이해하기 쉬움
- OpenCV에서는 윈도우(window, 창)가 활성화된 상태에서만 마우스나 키보드 이벤트 감지

모니터 스크린 상의 좌표

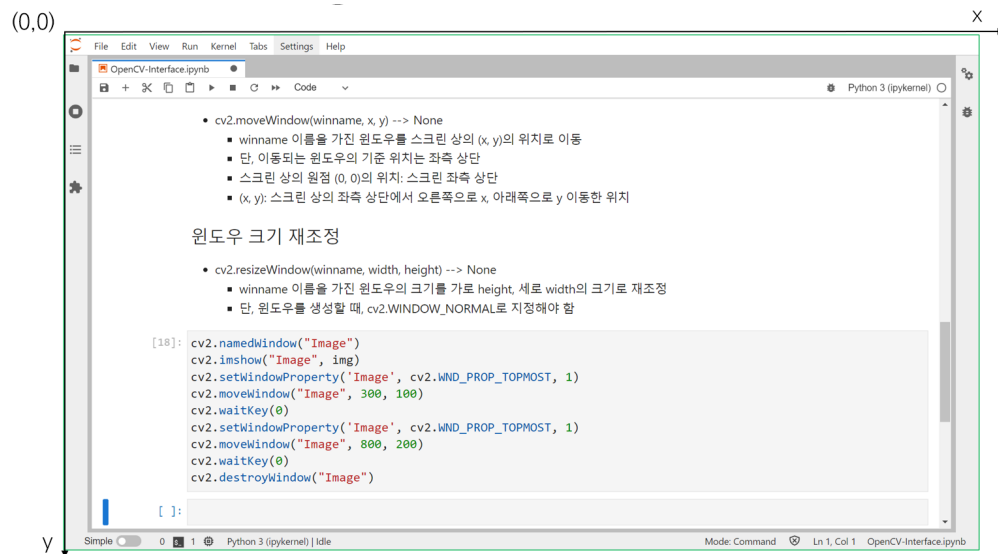


image.png

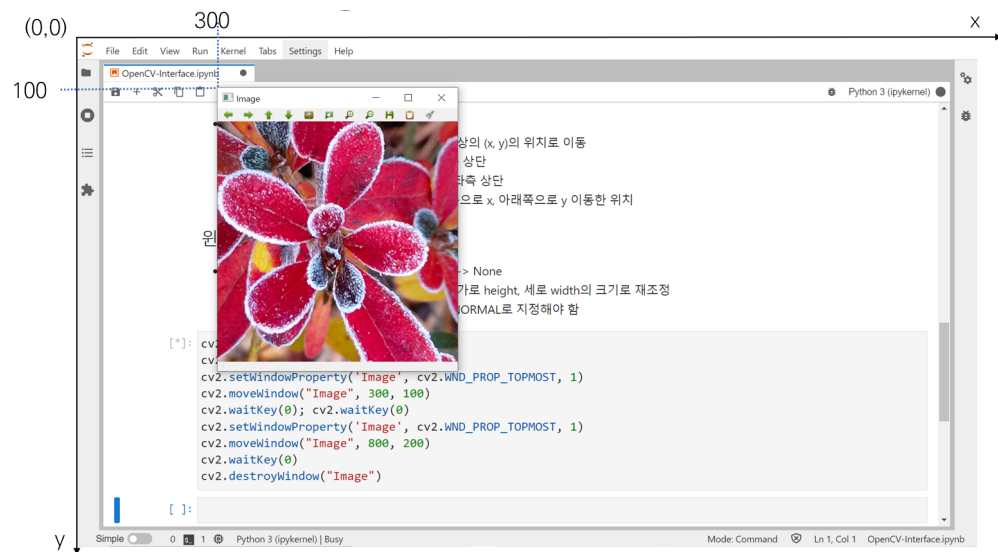


image.png

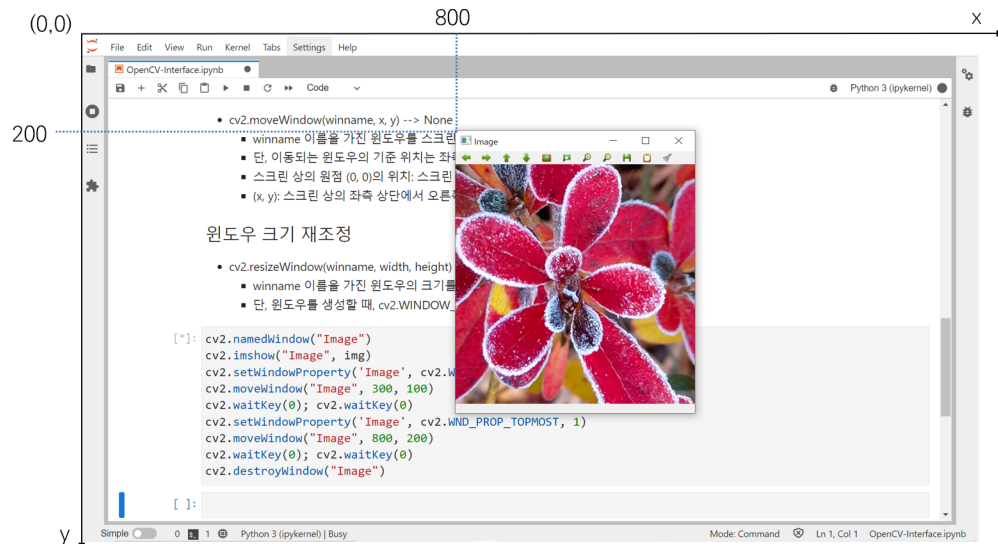


image.png

윈도우 이동

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

image = np.zeros((200, 400), np.uint8)
image[:] = 200                                     # 흰색 바탕

title1, title2 = 'Position1', 'Position2'           # 윈도우 이름
cv2.namedWindow(title1, cv2.WINDOW_AUTOSIZE)
cv2.namedWindow(title2)
cv2.moveWindow(title1, 150, 150)                     # 윈도우 이동
cv2.moveWindow(title2, 400, 50)

cv2.imshow(title1, image)                             # 행렬 원소를 영상으로 표시
cv2.imshow(title2, image)

cv2.waitKey(0)                                         # 키 이벤트(key event) 대기
cv2.destroyAllWindows()
```

윈도우 크기 변경

```
[5]: image = np.zeros((200, 300), np.uint8)
image.fill(255)                                # 모든 원소에 255(흰색) 지정

title1, title2 = 'AUTOSIZE', 'NORMAL'          # 윈도우 이름
cv2.namedWindow(title1, cv2.WINDOW_AUTOSIZE)
cv2.namedWindow(title2, cv2.WINDOW_NORMAL)

cv2.imshow(title1, image)
cv2.imshow(title2, image)
cv2.resizeWindow(title1, 400, 300)
cv2.resizeWindow(title2, 400, 300)
cv2.waitKey(0)
cv2.destroyAllWindows()                        # 열린 모든 윈도우 제거
```

1.5 이벤트 처리 함수

- 이벤트: 프로그램이 감지하고 처리할 수 있는 동작이나 사건
 - 사용자가 키보드의 키를 누르는 이벤트
 - 마우스를 움직이거나 마우스 버튼을 누르는 이벤트
 - 타이머(timer)나 통신 포트와 같은 하드웨어 장치가 발생시키는 이벤트
 - 사용자가 자체적으로 정의하는 이벤트
- 일반적으로 이벤트를 처리하기 위해 콜백(callback) 함수 사용
 - 개발자가 시스템 함수를 직접 호출하는 방식
 - 이벤트가 발생하거나 특정 시점에 도달했을 때 시스템이 개발자가 등록한 함수 호출
- OpenCV에서도 기본적인 이벤트 처리 함수 지원
 - 키보드 이벤트, 마우스 이벤트, 트랙바(trackbar) 이벤트

키보드 이벤트 제어

```
[8]: import numpy as np
import cv2

## switch case문을 사전(dictionary)으로 구현
```

```

switch_case = {
    ord('a'): "a키 입력",          # ord() 함수- 문자를 아
스키코드로 변환
    ord('b'): "b키 입력",
    0x41: "A키 입력",
    int('0x42', 16): "B키 입력",    # 16진수인 0x42를 10진수로
변환하면 66임
    2424832: "왼쪽 화살표키 입력",  # 0x250000
    2490368: "윗쪽 화살표키 입력",  # 0x260000
    2555904: "오른쪽 화살표키 입력", # 0x270000
    2621440: "아래쪽 화살표키 입력" # 0x280000
}

#image = np.ones((200, 300), np.float)      # 화소값이 1인 행렬 생성
image = np.ones((200, 300), float)          # 화소값이 1인 행렬 생성
cv2.namedWindow('Keyboard Event')           # 윈도우 이름
cv2.imshow('Keyboard Event', image)

while True:
    ↪                                         # 무
한 반복
    key = cv2.waitKeyEx(100)                # 100ms 동안 키 이벤트 대기
    if key == 27: break                      # ESC 키 누르면 종료

    try:
        result = switch_case[key]
        print(result)
    except KeyError:
        result = -1

cv2.destroyAllWindows()                    # 열린 모든 윈도우 제거

```

마우스 이벤트 제어

```
[9]: import numpy as np
import cv2

# 콜백 함수 : event값에 따른 마우스 버튼 종류 구분
def onMouse(event, x, y, flags, param):
    if event == cv2.EVENT_LBUTTONDOWN:
        print("마우스 왼쪽 버튼 누르기")
    elif event == cv2.EVENT_RBUTTONDOWN:
        print("마우스 오른쪽 버튼 누르기")
    elif event == cv2.EVENT_RBUTTONUP:
        print("마우스 오른쪽 버튼 떼기")
    elif event == cv2.EVENT_LBUTTONDBLCLK:
        print("마우스 왼쪽 버튼 더블클릭")

image = np.full((200, 300), 255, np.uint8) # 영상 생성

title1, title2 = "Mouse Event1", "Mouse Event2" # 윈도우 이름
cv2.imshow(title1, image) # 영상 보기
cv2.imshow(title2, image)

cv2.setMouseCallback('Mouse Event1', onMouse) # 마우스 콜백 함수
cv2.waitKey(0) # 키 이벤트 대기
cv2.destroyAllWindows() # 열린 모든 윈도우 제거
```

마우스 왼쪽 버튼 누르기

마우스 오른쪽 버튼 누르기

마우스 오른쪽 버튼 떼기

마우스 오른쪽 버튼 누르기

마우스 오른쪽 버튼 떼기

마우스 왼쪽 버튼 누르기

마우스 왼쪽 버튼 누르기

마우스 왼쪽 버튼 누르기

마우스 왼쪽 버튼 누르기

마우스 왼쪽 버튼 누르기

마우스 왼쪽 버튼 더블클릭
마우스 왼쪽 버튼 누르기
마우스 왼쪽 버튼 더블클릭
마우스 왼쪽 버튼 누르기
마우스 왼쪽 버튼 더블클릭
마우스 왼쪽 버튼 누르기
마우스 왼쪽 버튼 누르기
마우스 왼쪽 버튼 더블클릭
마우스 왼쪽 버튼 누르기
마우스 왼쪽 버튼 더블클릭
마우스 오른쪽 버튼 누르기
마우스 오른쪽 버튼 떼기
마우스 오른쪽 버튼 떼기
마우스 오른쪽 버튼 누르기
마우스 오른쪽 버튼 떼기
마우스 오른쪽 버튼 떼기
마우스 오른쪽 버튼 누르기
마우스 오른쪽 버튼 떼기
마우스 오른쪽 버튼 누르기
마우스 오른쪽 버튼 떼기
마우스 오른쪽 버튼 누르기
마우스 오른쪽 버튼 떼기

[]: