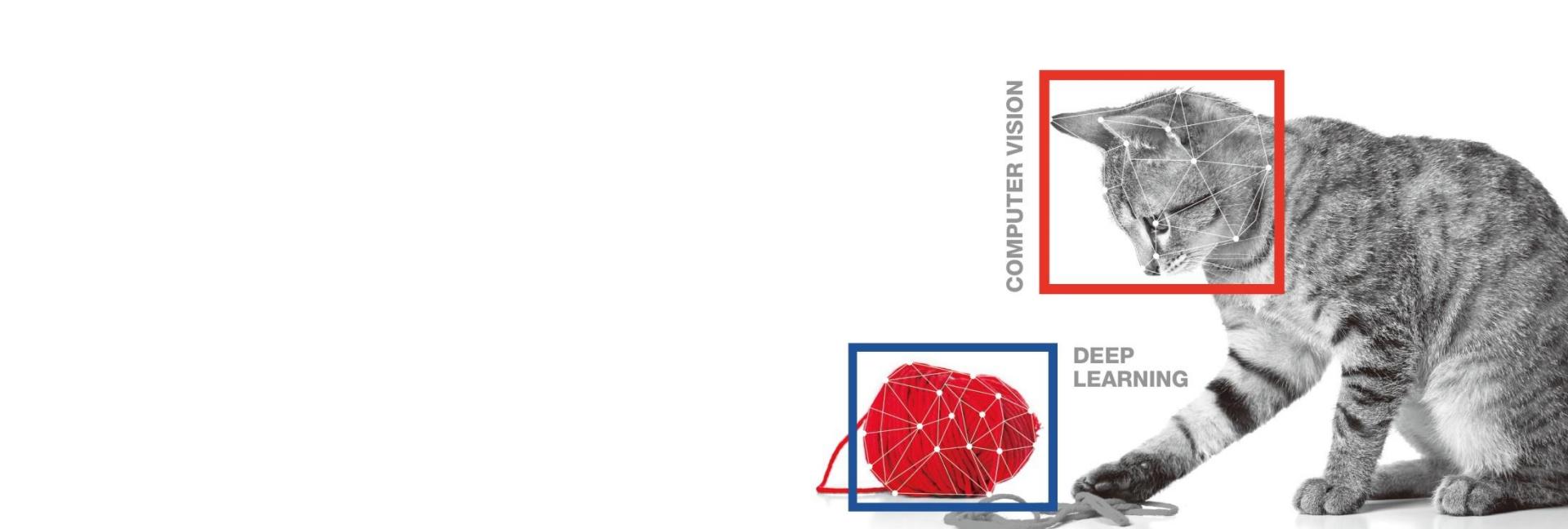


# 컴퓨터 비전과 딥러닝

## [강의교안 이용 안내]

- 본 강의교안의 저작권은 한빛아카데미(주)에 있습니다.
- 이 자료를 무단으로 전제하거나 배포할 경우 저작권법 136조에 의거하여 처벌을 받을 수 있습니다.



COMPUTER VISION

DEEP  
LEARNING

# 컴퓨터 비전과 딥러닝

## Chapter 02 OpenCV로 시작하는 컴퓨터 비전

# 차례

- 2.1 OpenCV 소개
- 2.2 프로그래밍 킥오프
- 2.3 객체지향 잘 활용하기
- 2.4 [프로그래밍 예제1] 영상을 읽고 표시하기
- 2.5 [프로그래밍 예제2] 영상 형태 변환하고 크기 축소하기
- 2.6 [프로그래밍 예제3] 웹 캠에서 비디오 읽기
- 2.7 [프로그래밍 예제4] 그래픽 기능과 사용자 인터페이스 만들기
- 2.8 [프로그래밍 예제5] 페인팅 기능 만들기

# Preview

## ■ 현대는 양호한 프로그래밍 환경

- 예전에는 알고리즘을 바닥부터 구현
- 현대는 함수 호출로 영상 처리하는 시대. 대표적 컴퓨터 비전 라이브러리는 OpenCV

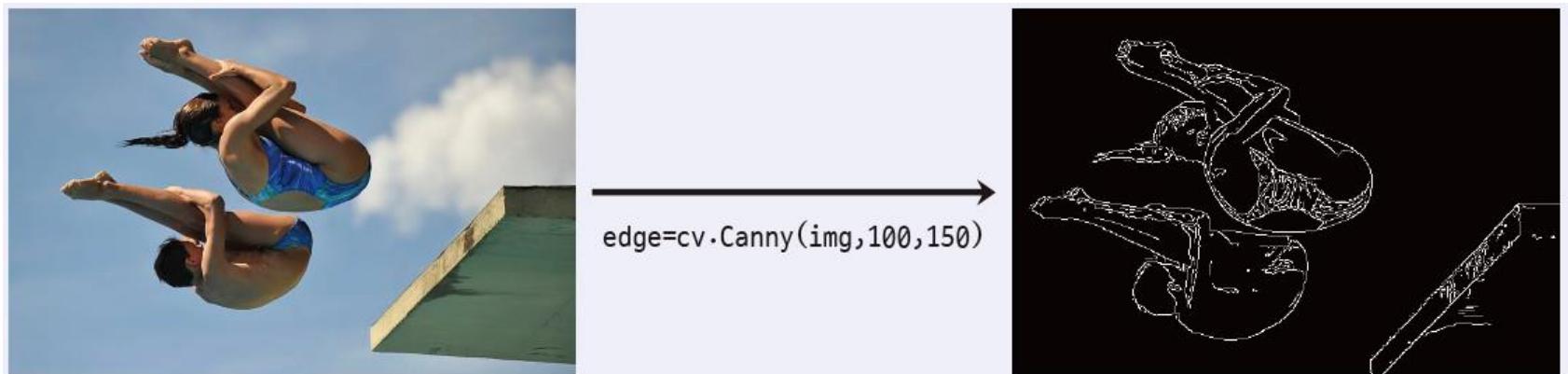


그림 2-1 에지 검출이라는 큰 일을 가뜬히 수행하는 OpenCV의 Canny 함수

## ■ 파이썬 언어와 OpenCV 라이브러리로 컴퓨터 비전 세계를 활짝~~~

## 2.1 OpenCV 소개

### ■ 인텔이 만들어 공개한 OpenCV

- 바퀴를 다시 발명 reinventing the wheel 하는 쓸데없는 노력을 방지할 목적
- 인텔 칩의 성능을 평가할 목적

# 개요와 간략한 역사

## ■ 개요

- 클래스와 함수는 C와 C++로 개발. 전체 코드는 180만 라인 이상
- 인터페이스 언어는 C, C++, 자바, 자바스크립트, 파이썬
- OS 플랫폼은 윈도우, 리눅스, macOS, 안드로이드, iOS
- 교차 플랫폼 지원
- 교육과 상업 목적 모두 무료

# 개요와 간략한 역사

## ■ 간략 역사

표 2-1 OpenCV의 역사

연도	사건
1998	<ul style="list-style-type: none"><li>인텔 직원인 개리 브라드스키(Gary Bradski)가 아이디어 제안</li></ul>
1999	<ul style="list-style-type: none"><li>오픈 소스로 공개하기로 결정하고 이름을 OpenCV로 정함</li></ul>
2000	<ul style="list-style-type: none"><li>CVPR 컨퍼런스에서 알파 버전 발표</li></ul>
2001-2005	<ul style="list-style-type: none"><li>5개의 베타 버전 발표</li></ul>
2005	<ul style="list-style-type: none"><li>스탠퍼드 대학교의 자율주행차인 스텐리의 개발 팀에 합류해 그랜드 챌린지 우승</li><li>OpenCV Korea 출범(<a href="https://cafe.naver.com/opencv">https://cafe.naver.com/opencv</a>)</li></ul>

# 개요와 간략한 역사

## ■ 간략 역사 (...계속)

2006	<ul style="list-style-type: none"><li>• OpenCV 1.0(C 인터페이스) 공개</li><li>•  로고 완성</li></ul>
2009	<ul style="list-style-type: none"><li>• OpenCV 2.0(C++ 인터페이스) 공개</li><li>• 파이썬과 자바 인터페이스 지원</li></ul>
2012	<ul style="list-style-type: none"><li>• 안드로이드와 iOS 지원 시작</li><li>• 깃허브로 마이그레이션</li></ul>
2015	<ul style="list-style-type: none"><li>• OpenCV 3.0 공개</li></ul>
2016	<ul style="list-style-type: none"><li>• 자바스크립트 인터페이스 지원 시작</li><li>• 딥러닝을 지원하는 DNN 모듈 추가</li></ul>
2018	<ul style="list-style-type: none"><li>• OpenCV 4.0 공개</li><li>• 고속 처리를 지원하는 OpenVINO 공개</li></ul>
2020	<ul style="list-style-type: none"><li>• Computer Vision and Deep Learning 코스 개설</li><li>• 전용 보드인 OpenCV AI Kit 출시</li></ul>
2022	<ul style="list-style-type: none"><li>• OpenCV 4.6 공개</li></ul>

# 공식 사이트

## ■ OpenCV를 지원하는 사이트

- 공식 홈페이지(<https://opencv.org>)
- 매뉴얼 사이트: 프로그래밍할 때 가장 많은 도움(<https://docs.opencv.org>)
  - ... 다음 슬라이드
- 깃허브
- 대한민국 OpenCV 사이트(<https://cafe.naver.com/opencv>)

# 공식 사이트

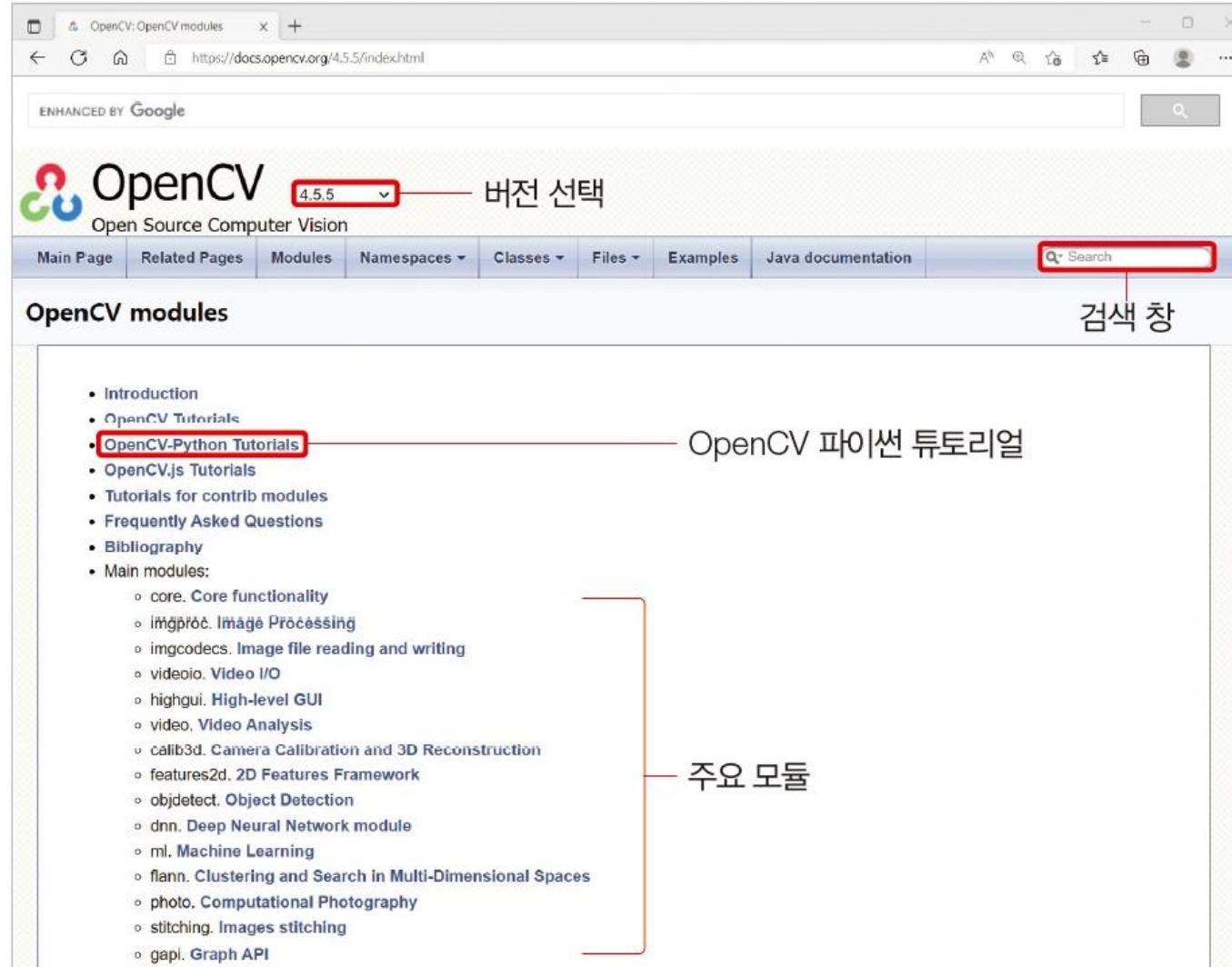


그림 2-2 OpenCV 매뉴얼 사이트

## 2.2 프로그래밍 킥오프

---

### ■ 프로그래밍 환경 갖추기

- 10~20분이면 필요한 소프트웨어 설치
- 모두 무료

# 파이썬 설치하기

## ■ 세 가지 소프트웨어 설치 필요함

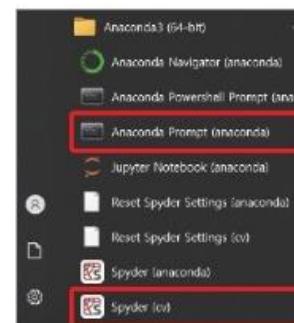
- 파이썬 컴파일러
- 통합 개발 환경(IDE)
- 라이브러리

## ■ 보통 아나콘다를 활용하여 설치

- 세 가지 소프트웨어를 한꺼번에 설치해줌
- 가상 환경 virtual environment 제공
  - 버전 충돌을 피하기 위해 새로운 프로젝트는 새로운 가상 환경에서 제작
- 이 책은 cv라는 가상 환경에서 프로그래밍 수행



(a) 아나콘다 다운로드 화면



(b) 윈도우 [시작] 버튼을 클릭해 아나콘다 설치 확인

그림 2-3 아나콘다와 스파이더 설치

# 파이썬 설치하기

## ■ 설치 절차

1. 아나콘다 설치하기
2. 가상 환경 생성하기
3. 가상 환경에서 IDE와 라이브러리 설치하기

```
(base) C:/> conda create -n cv
(base) C:/> conda activate cv
(cv) C:/> conda install spyder
(cv) C:/> pip install opencv-python
(cv) C:/> pip install tensorflow
(cv) C:/> conda list
```

❶ cv 가상 환경 생성  
❷ cv 가상 환경으로 이동  
❸ cv 가상 환경에 스파이더 설치  
❹ cv 가상 환경에 OpenCV 설치  
❺ cv 가상 환경에 텐서플로 설치  
❻ cv 가상 환경에 설치된 목록 보기

```
(cv) C:\Users\visor>
# packages in environment at C:\Users\visor\anaconda3\envs\cv:
#
# Name           Version        Build
absl-py          1.0.0          py39hd77b12b_6
alabaster       0.7.12         pyhd3eb1b0_0
appdirs          1.4.4          pyhd3eb1b0_0
numpy            1.22.1         pypi_0
numpydoc         1.1.0          pyhd3eb1b0_1
oauthlib         3.1.1          pypi_0
opencv-python    4.5.5.62       pypi_0
pyqt              5.9.2          py39hd77b12b_6
pyrsistent       0.18.0         py39h196d8e1_0
pysocks          1.7.1          py39haa95532_0
python            3.9.7          h6244533_1
spyder           5.1.5          py39haa95532_1
spyder-kernels   2.1.3          py39haa95532_0
sqlite            3.37.0         h2bbff1b_0
tensorboard      2.8.0          pypi_0
tensorboard-data-server 0.6.1          pypi_0
tensorboard-plugin-wit 1.8.1          pypi_0
tensorflow        2.7.0          pypi_0
```

그림 2-4 cv 가상 환경에 설치된 프로그램과 라이브러리 목록

# 스파이더 통합 개발 환경에서 프로그래밍하기

## ■ 스파이더 사용법

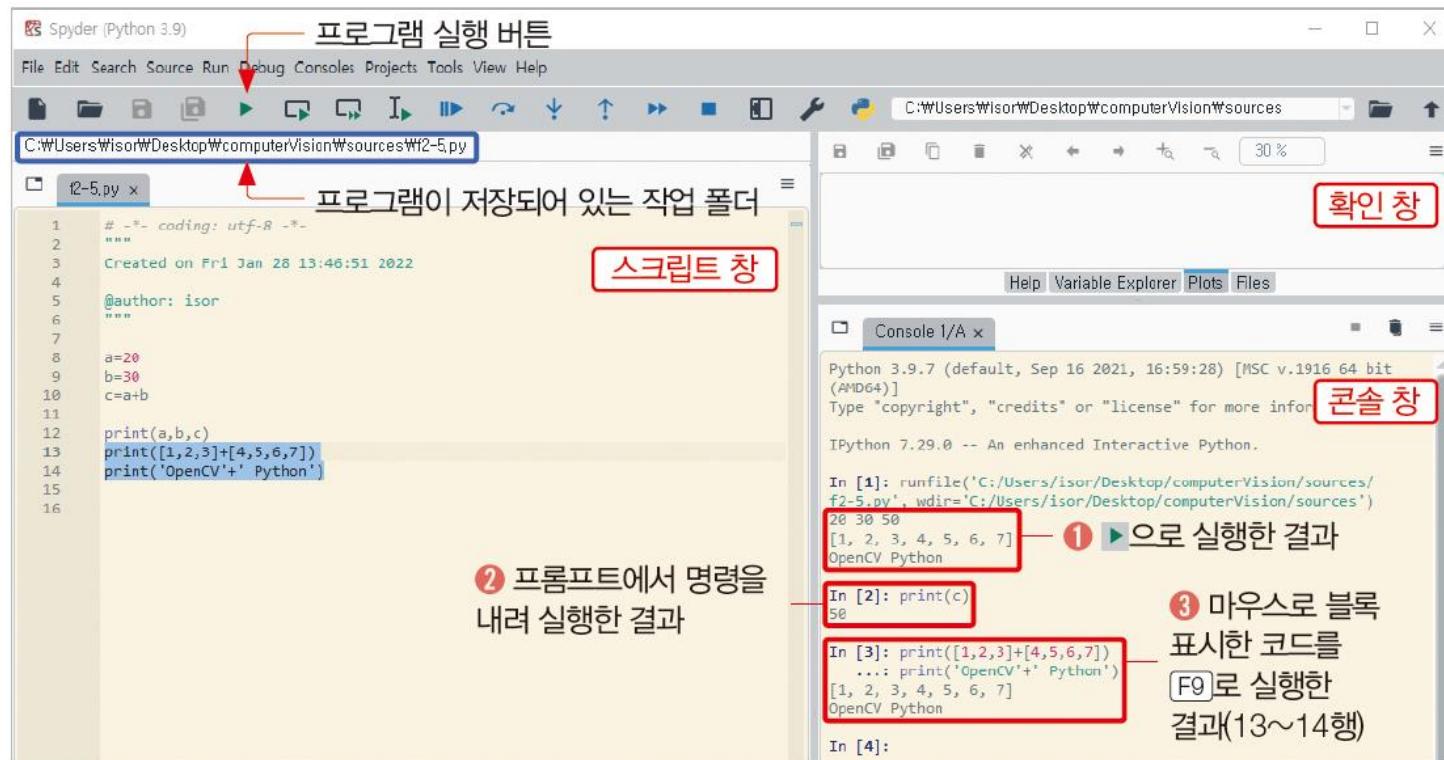


그림 2-5 스파이더 화면

- 파이썬은 인터프리터 방식이라 실행 끝난 뒤에 변수 값이 유지됨
  - 실행 후에 변수 값 확인 가능
  - F9 키를 이용한 부분 코드 실행이 가능 ← 컴퓨터 비전 프로그래밍에 크게 도움

# OpenCV 매뉴얼 활용하기

## ■ OpenCV-Python 튜토리얼을 잘 살필 것

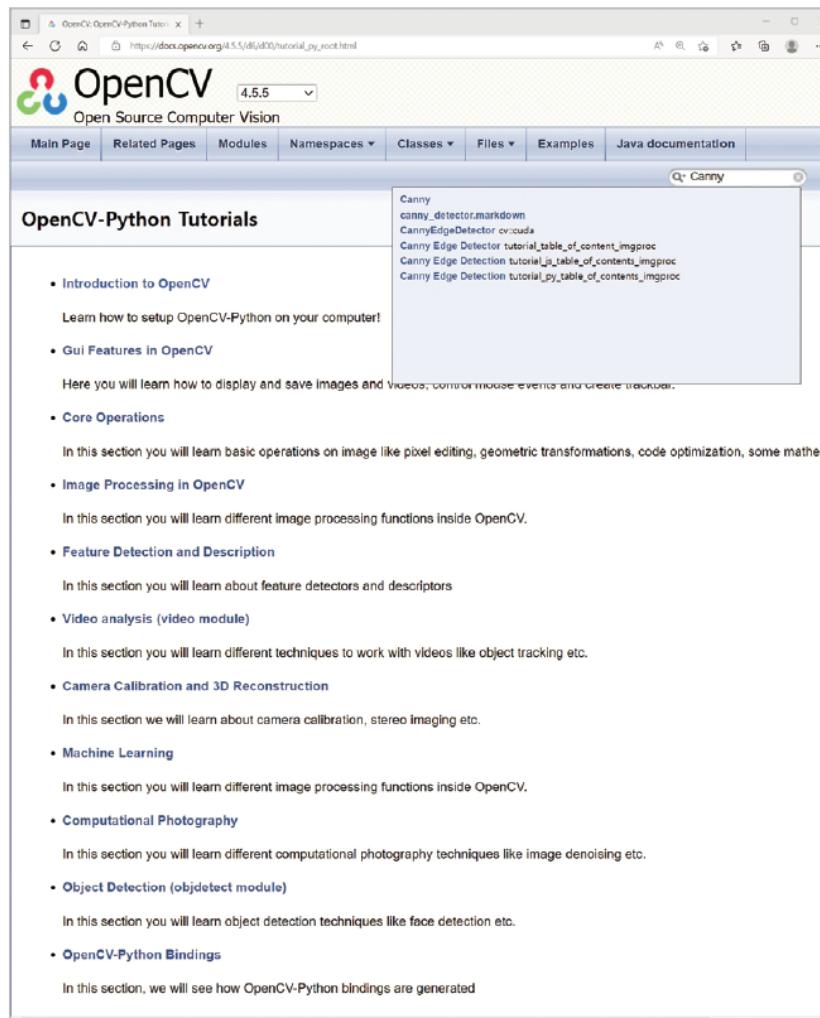
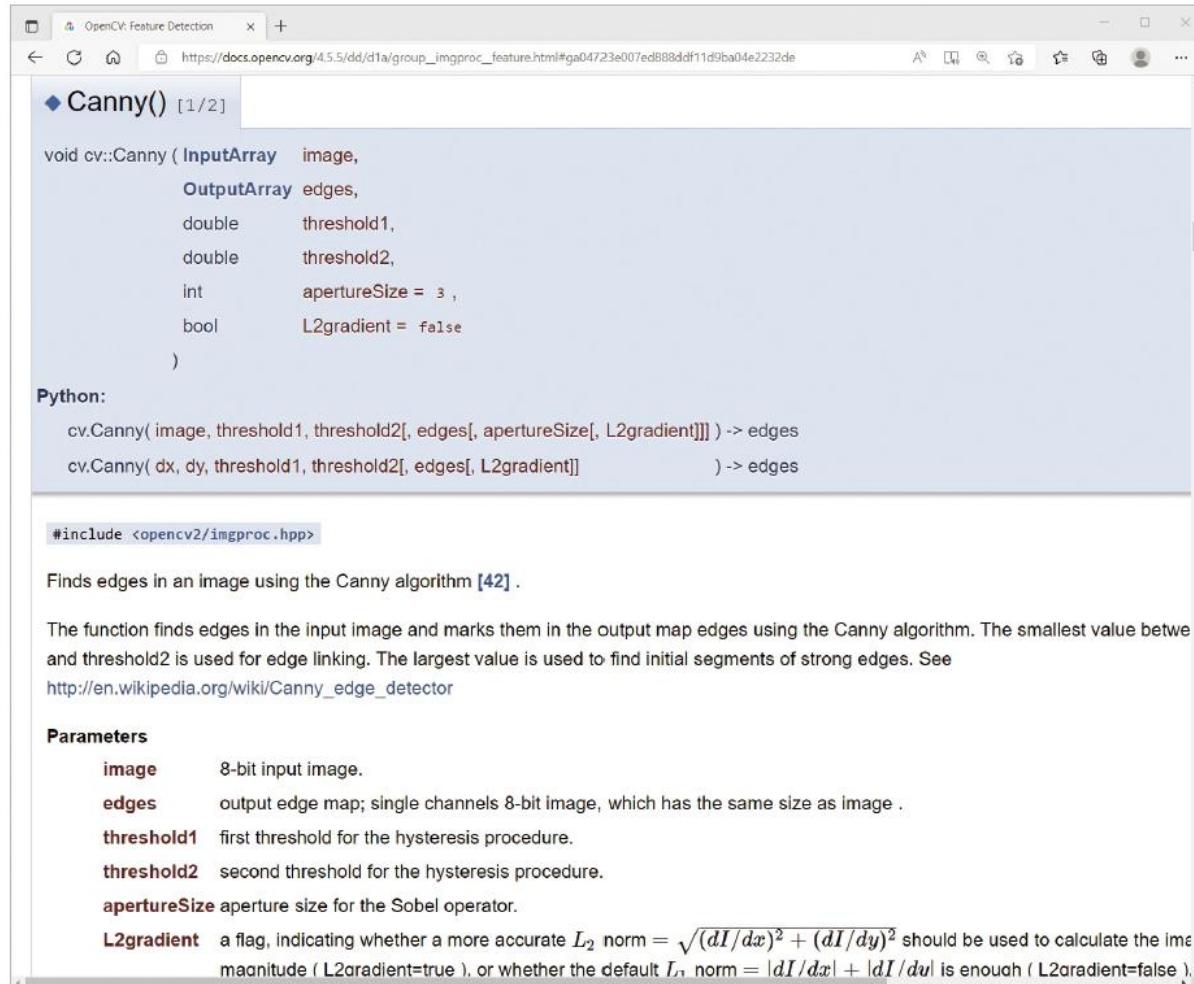


그림 2-6 OpenCV-Python 튜토리얼 화면

# OpenCV 매뉴얼 활용하기

## ■ 함수 선언을 잘 살필 것



The screenshot shows a web browser window displaying the OpenCV Feature Detection documentation. The main content is about the **Canny()** function. It includes code examples in C++ and Python, a detailed description of the function's purpose and parameters, and a list of parameters with their descriptions.

**C++:**

```
void cv::Canny ( InputArray image,
                 OutputArray edges,
                 double threshold1,
                 double threshold2,
                 int apertureSize = 3,
                 bool L2gradient = false
               )
```

**Python:**

```
cv.Canny( image, threshold1, threshold2[, edges[, apertureSize[, L2gradient]]] ) -> edges
cv.Canny( dx, dy, threshold1, threshold2[, edges[, L2gradient]] ) -> edges
```

**#include <opencv2/imgproc.hpp>**

Finds edges in an image using the Canny algorithm [42].

The function finds edges in the input image and marks them in the output map edges using the Canny algorithm. The smallest value between threshold1 and threshold2 is used for edge linking. The largest value is used to find initial segments of strong edges. See [http://en.wikipedia.org/wiki/Canny\\_edge\\_detector](http://en.wikipedia.org/wiki/Canny_edge_detector)

**Parameters**

- image** 8-bit input image.
- edges** output edge map; single channels 8-bit image, which has the same size as image .
- threshold1** first threshold for the hysteresis procedure.
- threshold2** second threshold for the hysteresis procedure.
- apertureSize** aperture size for the Sobel operator.
- L2gradient** a flag, indicating whether a more accurate  $L_2$  norm =  $\sqrt{(dI/dx)^2 + (dI/dy)^2}$  should be used to calculate the image magnitude ( L2gradient=true ), or whether the default  $L_1$  norm =  $|dI/dx| + |dI/dy|$  is enough ( L2gradient=false ).

그림 2-7 OpenCV 함수 선언-Canny 함수 사례

## 2.3 객체지향 잘 활용하기

---

### ■ 파이썬은 객체지향 언어

- 객체지향은 컴퓨터 비전 프로그래밍에 매우 유리

# 객체 다루기

## ■ [프로그램 2-1]은 객체지향 특성을 설명

### 프로그램 2-1 numpy.ndarray 클래스 형의 객체를 만들고 멤버 함수 적용하기

```
01 import numpy as np
02
03 a=np.array([4,5,0,1,2,3,6,7,8,9,10,11])
04 print(a) ①
05 print(type(a)) ②
06 print(a.shape) ③
07 a.sort()
08 print(a) ④
09
10 b=np.array([-4.3,-2.3,12.9,8.99,10.1,-1.2])
11 b.sort()
12 print(b) ⑤
13
14 c=np.array(['one','two','three','four','five','six','seven'])
15 c.sort()
16 print(c) ⑥
```

```
[ 4  5  0  1  2  3  6  7  8  9 10 11]          ①
<class 'numpy.ndarray'>                      ②
(12,)                                         ③
[ 0  1  2  3  4  5  6  7  8  9 10 11]          ④
[-4.3 -2.3 -1.2  8.99 10.1 12.9 ]          ⑤
['five' 'four' 'one' 'seven' 'six' 'three' 'two'] ⑥
```

# 객체 다루기

## ■ 객체지향 특성과 강점

- 객체는 능동적: 자신이 소유한 멤버 함수를 능동적으로 호출
- 필요한 만큼 얼마든지 찍어낼 수 있음



(a) a 객체



(b) b 객체



(c) c 객체

그림 2-8 np.ndarray 클래스로 찍어낸 객체들

# 객체 확인하기 - type과 dir 내장 함수

## ■ 객체의 클래스를 알려주는 type과 사용 가능한 멤버 함수를 알려주는 dir

```
In [1]: type(a)
       numpy.ndarray
In [2]: dir(a)
       ['T', '__abs__', '__add__', '__and__', ..., 'all', 'any', ..., 'max', ..., 'shape', ..., 'sort', ...]
```

## ■ 함수가 하는 일을 알아내려면 help

```
In [3]: 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.
```

## 2.4 [프로그래밍 예제1] 영상을 읽고 표시하기

### ■ 처음 해보는 OpenCV 프로그래밍

프로그램 2-2 영상 파일을 읽고 원도우에 디스플레이하기

```
01 import cv2 as cv
02 import sys
03
04 img=cv.imread('soccer.jpg')      # 영상 읽기
05
06 if img is None:
07     sys.exit('파일을 찾을 수 없습니다.')
08
09 cv.imshow('Image Display',img)  # 원도우에 영상 표시
10
11 cv.waitKey()
12 cv.destroyAllWindows()
```



# OpenCV에서 영상은 numpy.ndarray 클래스 형의 객체

## ■ numpy는 다차원 배열을 위한 사실상 표준 모듈

- 이런 이유로 OpenCV는 영상을 numpy.ndarray로 표현
- OpenCV가 다루는 영상은 numpy가 제공하는 다양한 기능(함수)을 사용할 수 있음

```
In [1]: type(img)
```

```
numpy.ndarray
```

```
In [2]: img.shape
```

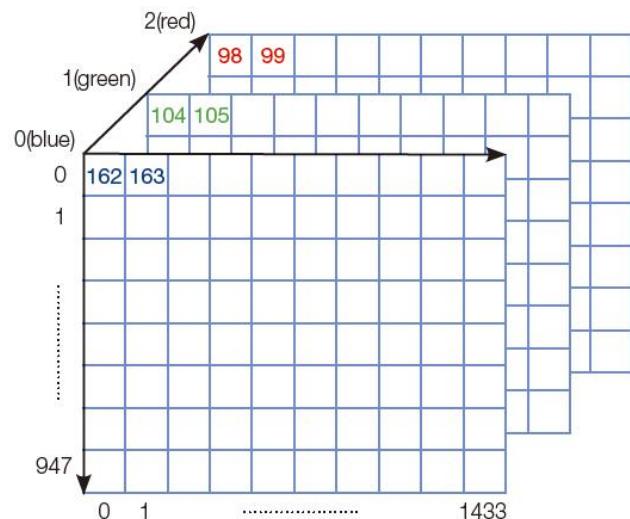
```
(948, 1434, 3)
```

# OpenCV에서 영상은 numpy.ndarray 클래스 형의 객체

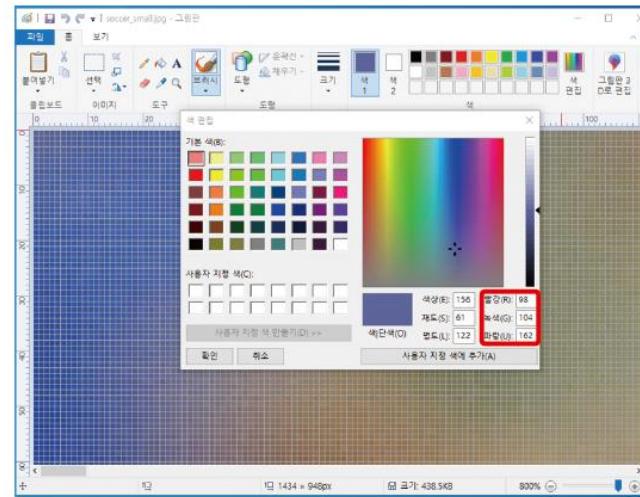
## ■ 영상의 표현

- 화소의 위치 (r,c) 또는 (y,x)
- 화소값 조사

```
In [3]: print(img[0,0,0], img[0,0,1], img[0,0,2]) # (0,0) 화소 조사  
162 104 98  
In [4]: print(img[0,1,0], img[0,1,1], img[0,1,2]) # (0,1) 화소 조사  
163 105 99
```



(a) 프로그램으로 조사



(b) 그림판으로 조사

그림 2-9 img 객체가 표현하는 영상의 구조와 내용

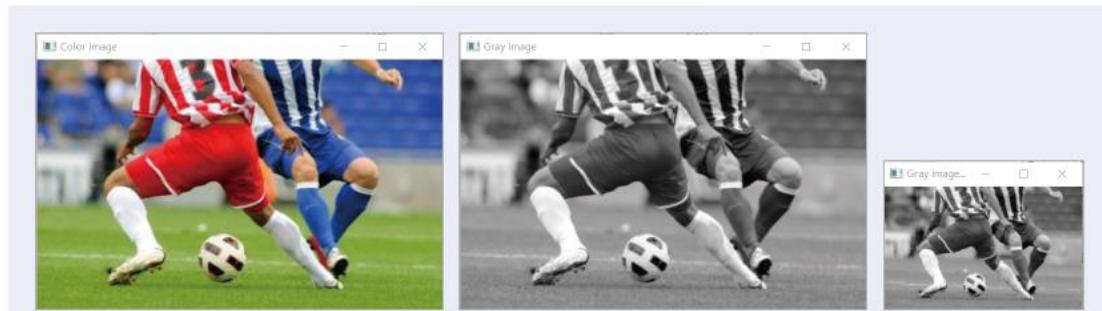
## 2.5 [프로그래밍 예제2] 영상 형태 변환하고 크기 축소하기

### ■ [프로그램 2-3]

프로그램 2-3

영상을 명암 영상으로 변환하고 반으로 축소하기

```
01 import cv2 as cv
02 import sys
03
04 img=cv.imread('soccer.jpg')
05
06 if img is None:
07     sys.exit('파일을 찾을 수 없습니다.')
08
09 gray=cv.cvtColor(img,cv.COLOR_BGR2GRAY)          # BGR 컬러 영상을 명암 영상으로 변환
10 gray_small=cv.resize(gray,dsize=(0,0),fx=0.5,fy=0.5) # 반으로 축소
11
12 cv.imwrite('soccer_gray.jpg',gray)                # 영상을 파일에 저장
13 cv.imwrite('soccer_gray_small.jpg',gray_small)
14
15 cv.imshow('Color image',img)
16 cv.imshow('Gray image',gray)
17 cv.imshow('Gray image small',gray_small)
18
19 cv.waitKey()
20 cv.destroyAllWindows()
```



## 2.5 [프로그래밍 예제2] 영상 형태 변환하고 크기 축소하기

### ■ cvtColor 함수가 컬러 영상을 명암 영상으로 바꾸는 방법

$$I = \text{round}(0.299 \times R + 0.587 \times G + 0.114 \times B) \quad (2.1)$$

(참조: [https://docs.opencv.org/3.4/de/d25/imgproc\\_color\\_conversions.html](https://docs.opencv.org/3.4/de/d25/imgproc_color_conversions.html))

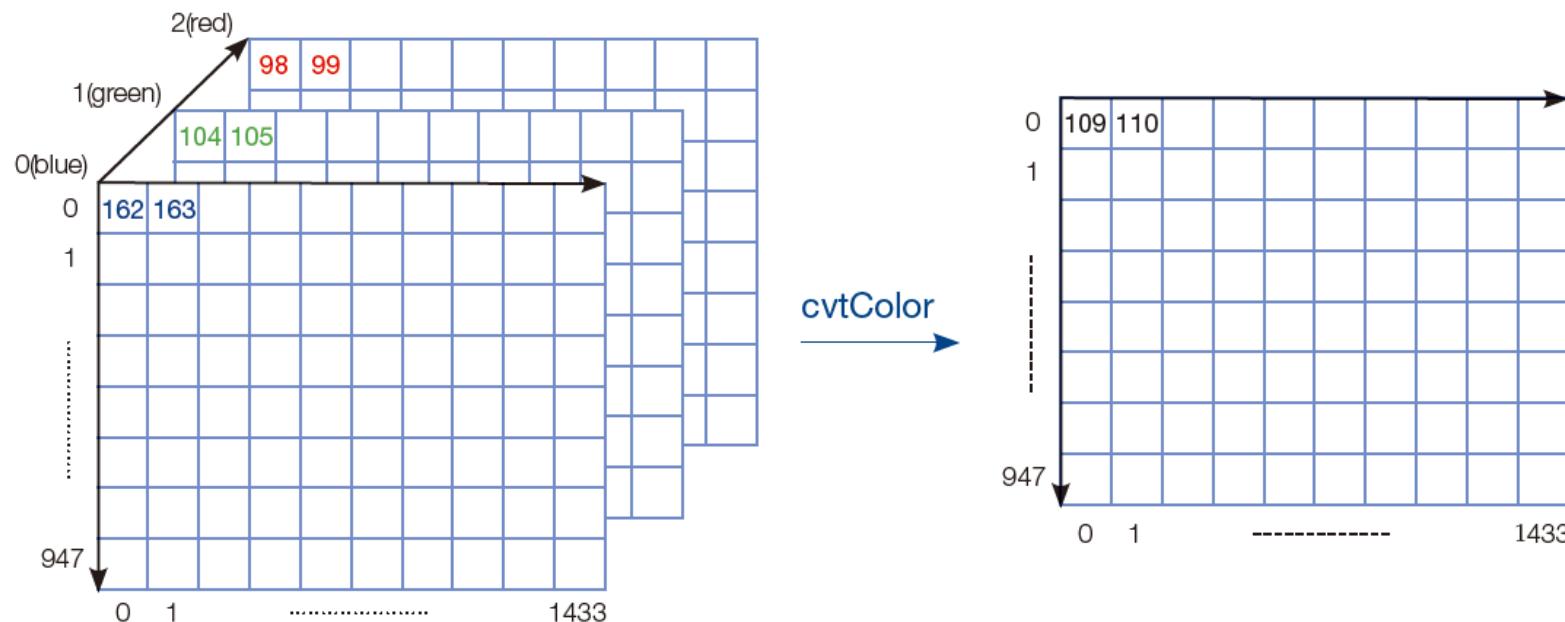


그림 2-10 BGR 컬러 영상을 명암 영상으로 변환

## 2.6 [프로그래밍 예제3] 웹 캠에서 비디오 읽기

### ■ 웹 캠에서 비디오 읽기

프로그램 2-4 웹 캠으로 비디오 획득하기

```
01 import cv2 as cv
02 import sys
03
04 cap=cv.VideoCapture(0, cv.CAP_DSHOW) # 카메라와 연결 시도
05
06 if not cap.isOpened():
07     sys.exit('카메라 연결 실패')
08
09 while True:
10     ret,frame=cap.read()           # 비디오를 구성하는 프레임 획득
11
12     if not ret:
13         print('프레임 획득에 실패하여 루프를 나갑니다.')
14         break
15
16     cv.imshow('Video display',frame)
17
18     key=cv.waitKey(1)             # 1밀리초 동안 키보드 입력 기다림
19     if key==ord('q'):            # 'q' 키가 들어오면 루프를 빠져나감
20         break
21
22     cap.release()                # 카메라와 연결을 끊음
23     cv.destroyAllWindows()
```



## 2.6 [프로그래밍 예제3] 웹 캠에서 비디오 읽기

### ■ 비디오에서 영상 수집하기(numpy의 hstack 함수 사용)

프로그램 2-5

비디오에서 수집한 영상을 이어 붙이기

```
01 import cv2 as cv
02 import numpy as np
03 import sys
04
05 cap=cv.VideoCapture(0,cv.CAP_DSHOW)      # 카메라와 연결 시도
06
07 if not cap.isOpened():
08     sys.exit('카메라 연결 실패')
09
10 frames=[]
11 while True:
12     ret,frame=cap.read()                  # 비디오를 구성하는 프레임 획득
13
14     if not ret:
15         print('프레임 획득에 실패하여 루프를 나갑니다.')
16         break
17
```

## 2.6 [프로그래밍 예제3] 웹 캠에서 비디오 읽기

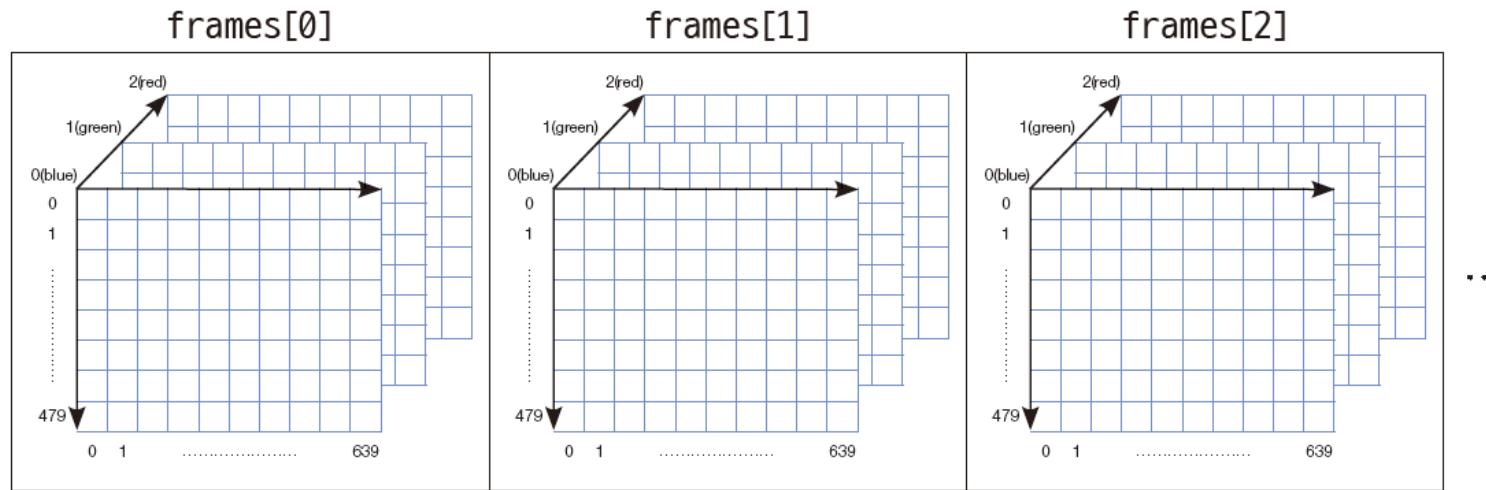
```
18     cv.imshow('Video display',frame)
19
20     key=cv.waitKey(1)                      # 1밀리초 동안 키보드 입력 기다림
21     if key==ord('c'):                     # 'c' 키가 들어오면 프레임을 리스트에 추가
22         frames.append(frame)
23     elif key==ord('q'):                   # 'q' 키가 들어오면 루프를 빠져나감
24         break
25
26 cap.release()                           # 카메라와 연결을 끊음
27 cv.destroyAllWindows()
28
29 if len(frames)>0:                      # 수집된 영상이 있으면
30     imgs=frames[0]
31     for i in range(1,min(3,len(frames))): # 최대 3개까지 이어 붙임
32         imgs=np.hstack((imgs,frames[i]))
33
34 cv.imshow('collected images',imgs)
35
36 cv.waitKey()
37 cv.destroyAllWindows()
```

## 2.6 [프로그래밍 예제3] 웹 캠에서 비디오 읽기

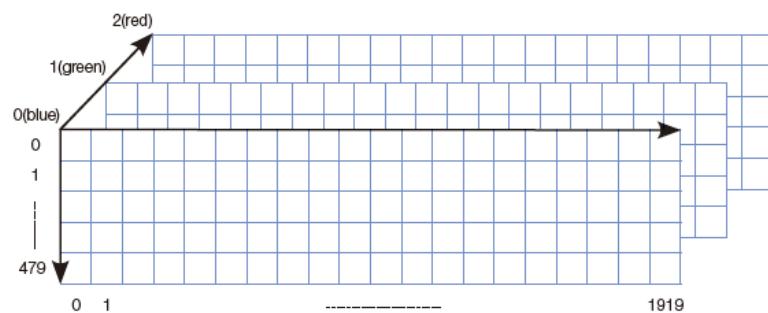


## 2.6 [프로그래밍 예제3] 웹 캠에서 비디오 읽기

### ■ [프로그램 2-5]의 자료구조



(a) frames 리스트



(b) imgs 배열

```
In [1]: len(frames)
```

```
7
```

```
In [2]: frames[0].shape
```

```
(480, 640, 3)
```

```
In [3]: type(imgs)
```

```
numpy.ndarray
```

```
In [4]: imgs.shape
```

```
(480, 1920, 3)
```

그림 2-11 [프로그램 2-5]의 자료 구조

## 2.7 [프로그래밍 예제4] 그래픽 기능과 사용자 인터페이스 만들기

### ■ OpenCV의 그래픽 기능

- 영상에 글씨나 도형을 넣는데 유용([그림 2-6]의 Gui features in OpenCV 참조)
- line, rectangle, polylines, circle, ellipse, putText 함수

## 2.7 [프로그래밍 예제4] 그래픽 기능과 사용자 인터페이스 만들기

### ■ [프로그램 2-6] 영상에 도형을 그리고 글씨 쓰기

프로그램 2-6 영상에 도형을 그리고 글자 쓰기

```
01 import cv2 as cv
02 import sys
03
04 img=cv.imread('girl_laughing.jpg')
05
06 if img is None:
07     sys.exit('파일을 찾을 수 없습니다.')
08
09 cv.rectangle(img,(830,30),(1000,200),(0,0,255),2) # 직사각형 그리기
10 cv.putText(img,'laugh',(830,24),cv.FONT_HERSHEY_SIMPLEX,1,(255,0,0),2) # 글씨 쓰기
11
12 cv.imshow('Draw',img)
13
14 cv.waitKey()
15 cv.destroyAllWindows()
```



red

굵기

blue

## 2.7 [프로그래밍 예제4] 그래픽 기능과 사용자 인터페이스 만들기



## 2.7 [프로그래밍 예제4] 그래픽 기능과 사용자 인터페이스 만들기

### ■ 함수 선언에 대한 이해

#### ◆ rectangle() [1/2]

```
void cv::rectangle( InputOutputArray img,
                    Point pt1,
                    Point pt2,
                    const Scalar & color,
                    int thickness = 1 ,
                    int lineType = LINE_8 ,
                    int shift = 0
                )
```

#### Python:

```
cv.rectangle( img, pt1, pt2, color[, thickness[, lineType[, shift]]] ) -> img
cv.rectangle( img, rec, color[, thickness[, lineType[, shift]]] ) -> img
```

```
#include <opencv2/imgproc.hpp>
```

Draws a simple, thick, or filled up-right rectangle.

The function `cv::rectangle` draws a rectangle outline or a filled rectangle whose two opposite corners are `pt1` and `pt2`.

#### Parameters

- `img` Image.
- `pt1` Vertex of the rectangle.
- `pt2` Vertex of the rectangle opposite to `pt1`.
- `color` Rectangle color or brightness (grayscale image).
- `thickness` Thickness of lines that make up the rectangle. Negative values, like FILLED, mean that the function has to draw a filled rectangle.
- `lineType` Type of the line. See LineTypes
- `shift` Number of fractional bits in the point coordinates.

그림 2-12 OpenCV 공식 사이트가 제공하는 `rectangle` 함수의 선언([https://docs.opencv.org/4.x/d6/d6e/group\\_\\_imgproc\\_\\_draw.html#ga07d2f74cadcf8e305e810ce8eed13bc9](https://docs.opencv.org/4.x/d6/d6e/group__imgproc__draw.html#ga07d2f74cadcf8e305e810ce8eed13bc9))

## 2.7 [프로그래밍 예제4] 그래픽 기능과 사용자 인터페이스 만들기

### ■ [프로그램 2-7] 마우스를 통한 상호작용(콜백 함수에 대한 이해 필요)

프로그램 2-7

마우스로 클릭한 곳에 직사각형 그리기

```
01 import cv2 as cv
02 import sys
03
04 img=cv.imread('girl_laughing.jpg')
05
06 if img is None:
07     sys.exit('파일을 찾을 수 없습니다.')
08
09 def draw(event,x,y,flags,param):      # 콜백 함수
10     if event==cv.EVENT_LBUTTONDOWN:    # 마우스 왼쪽 버튼 클릭했을 때
11         cv.rectangle(img,(x,y),(x+200,y+200),(0,0,255),2)
12     elif event==cv.EVENT_RBUTTONDOWN:  # 마우스 오른쪽 버튼 클릭했을 때
13         cv.rectangle(img,(x,y),(x+100,y+100),(255,0,0),2)
14
15     cv.imshow('Drawing',img)
16
17 cv.namedWindow('Drawing')
18 cv.imshow('Drawing',img)
19
20 cv.setMouseCallback('Drawing',draw)   # Drawing 윈도우에 draw 콜백 함수 지정
21
22 while(True):                      # 마우스 이벤트가 언제 발생할지 모르므로 무한 반복
23     if cv.waitKey(1)==ord('q'):
24         cv.destroyAllWindows()
25         break
```



## 2.7 [프로그래밍 예제4] 그래픽 기능과 사용자 인터페이스 만들기

### ■ [프로그램 2-8] 마우스 드래그로 도형 크기 조절하기

프로그램 2-8

마우스 드래그로 직사각형 그리기

```
01 import cv2 as cv
02 import sys
03
04 img=cv.imread('girl_laughing.jpg')
05
06 if img is None:
07     sys.exit('파일을 찾을 수 없습니다.')
08
09 def draw(event,x,y,flags,param):
10     global ix,iy
11
12     if event==cv.EVENT_LBUTTONDOWN:    # 마우스 왼쪽 버튼 클릭했을 때 초기 위치 저장
13         ix,iy=x,y
14     elif event==cv.EVENT_LBUTTONUP:   # 마우스 왼쪽 버튼 클릭했을 때
15         cv.rectangle(img,(ix,iy),(x,y),(0,0,255),2)
16
17     cv.imshow('Drawing',img)
18
19 cv.namedWindow('Drawing')
20 cv.imshow('Drawing',img)
21
22 cv.setMouseCallback('Drawing',draw)
23
24 while(True):
25     if cv.waitKey(1)==ord('q'):
26         cv.destroyAllWindows()
27         break
```



## 2.8 [프로그래밍 예제 5] 페인팅

### ■ [프로그램 2-9]: 페인팅 기능

프로그램 2-9

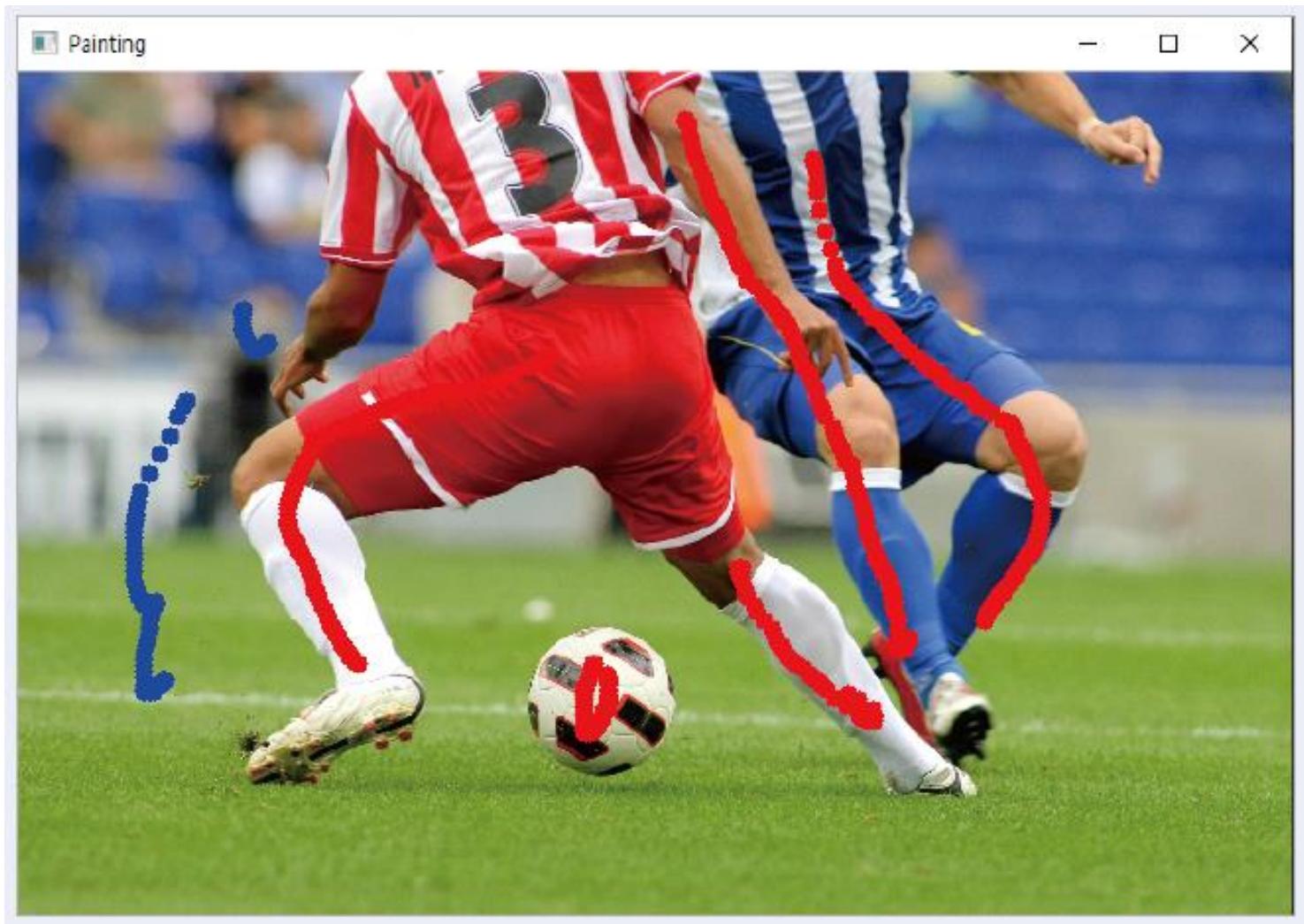
빨간색 붓과 파란색 붓으로 페인팅하기

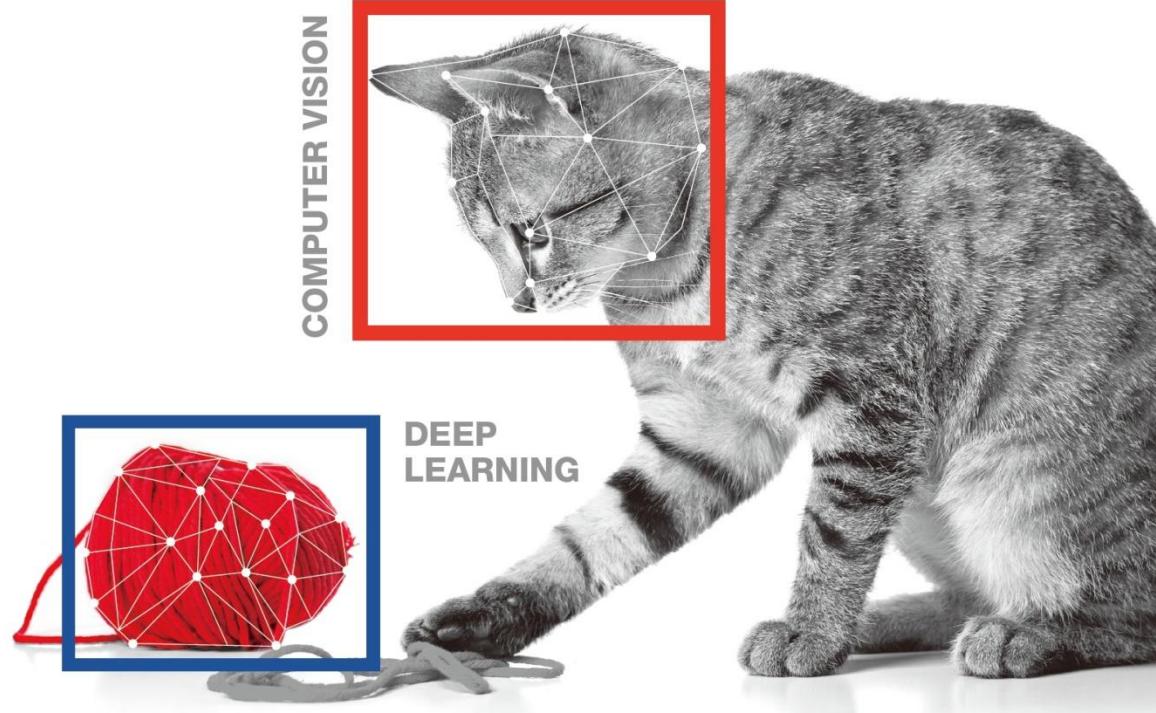
```
01 import cv2 as cv
02 import sys
03
04 img=cv.imread('soccer.jpg')
05
06 if img is None:
07     sys.exit('파일을 찾을 수 없습니다.')
08
09 BrushSiz=5                      # 붓의 크기
10 LColor,RColor=(255,0,0),(0,0,255)  # 파란색과 빨간색
```

## 2.8 [프로그래밍 예제 5] 페인팅

```
11
12 def painting(event,x,y,flags,param):
13     if event==cv.EVENT_LBUTTONDOWN:
14         cv.circle(img,(x,y),BrushSiz,LColor,-1)      # 마우스 왼쪽 버튼 클릭하면 파란색
15     elif event==cv.EVENT_RBUTTONDOWN:
16         cv.circle(img,(x,y),BrushSiz,RColor,-1)      # 마우스 오른쪽 버튼 클릭하면 빨간색
17     elif event==cv.EVENT_MOUSEMOVE and flags==cv.EVENT_FLAG_LBUTTON:
18         cv.circle(img,(x,y),BrushSiz,LColor,-1)      # 왼쪽 버튼 클릭하고 이동하면 파란색
19     elif event==cv.EVENT_MOUSEMOVE and flags==cv.EVENT_FLAG_RBUTTON:
20         cv.circle(img,(x,y),BrushSiz,RColor,-1)      # 오른쪽 버튼 클릭하고 이동하면 빨간색
21
22     cv.imshow('Painting',img)                      # 수정된 영상을 다시 그림
23
24 cv.namedWindow('Painting')
25 cv.imshow('Painting',img)
26
27 cv.setMouseCallback('Painting',painting)
28
29 while(True):
30     if cv.waitKey(1)==ord('q'):
31         cv.destroyAllWindows()
32         break
```

## 2.8 [프로그래밍 예제 5] 페인팅





# 컴퓨터 비전과 딥러닝

감사합니다.