

영상처리 실제

Image Processing Practice

신재혁

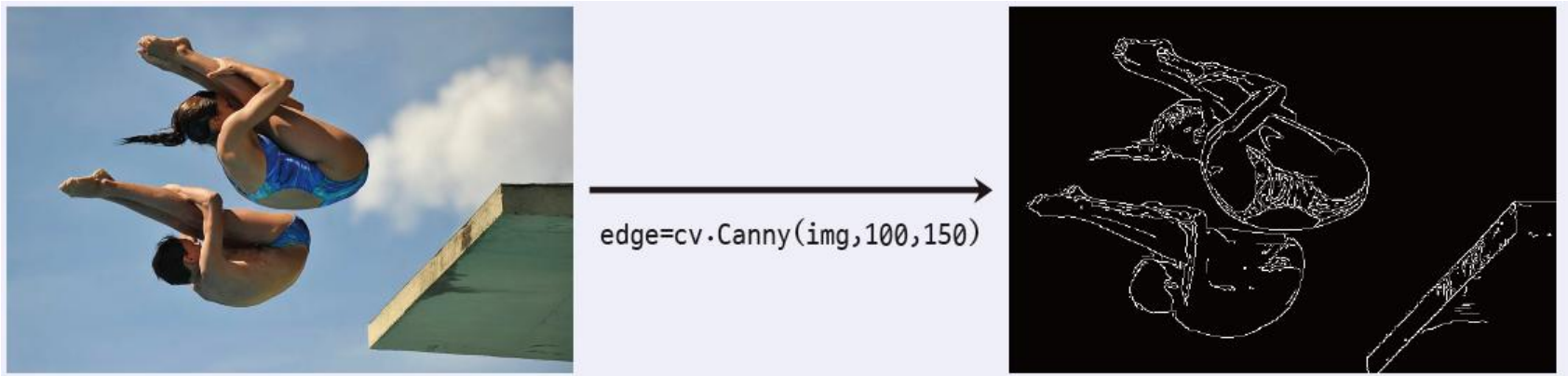
naezang@cbnu.ac.kr

2장 OpenCV 설치와 개요

OpenCV 소개

OpenCV 소개

- 현대의 프로그래밍 환경
 - 예전에는 알고리즘을 바닥부터 구현
 - 함수 호출로 영상 처리



- 대표적 컴퓨터 비전 라이브러리: OpenCV
- 현재는 이미지 입력 → 딥러닝이 영상 처리

OpenCV 소개

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

OpenCV 소개

- 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

OpenCV 소개

- OpenCV - Open Source Computer Vision Library
 - 영상 처리와 컴퓨터 비전 관련 오픈 소스 라이브러리
 - 2,500개가 넘는 알고리즘으로 구성
 - C, C++, 파이썬(Python), 매트랩 인터페이스 제공
 - 윈도우즈, 리눅스, 안드로이드, 맥 OS 등 다양한 운영체제 지원
 - CUDA와 OpenCL 인터페이스 개발
 - 인텔사에서 개발한 IPL (Image Processing Library) 기반
 - 2006년 V1.0 - C 기반 API
 - 2009년 V2.0 - C++ 기반 API
 - 2015년 V3.1 - 고급 영상처리 알고리즘 (Feature Extraction 등)
 - 2018년 V4.0 - 신경망/딥러닝 알고리즘

OpenCV 소개

- OpenCV의 간략한 역사

연도	사건
1998	• 인텔 직원인 개리 브라드스키(Gary Bradski)가 아이디어 제안
1999	• 오픈 소스로 공개하기로 결정하고 이름을 OpenCV로 정함
2000	• CVPR 컨퍼런스에서 알파 버전 발표
2001-2005	• 5개의 베타 버전 발표
2005	• 스탠퍼드 대학교의 자율주행차인 스탠리의 개발 팀에 합류해 그랜드 챌린지 우승 • OpenCV Korea 출범(https://cafe.naver.com/opencv)

OpenCV 소개

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

OpenCV 소개

〈표 2.1.1〉 OpneCV 버전별 특징

1.0 버전	2.0 버전	2.1 버전
<ul style="list-style-type: none"> • C 언어 기반 API • 구조체 기반 데이터 구조 사용 • 비주얼 스튜디오에서 라이브러리 컴파일 후 사용 • highgui 모듈에서 8비트 PNG, JPEG2000 입출력 지원 • 샘플 예제 파일 추가(calibrate.cpp, inpaint.cpp, leter_recog.cpp 등) 	<ul style="list-style-type: none"> • C++ 언어 기반 API • 클래스 기반 데이터 구조 도입 • CMake를 이용하여 라이브러리 컴파일 후 사용 가능 • highGUI에서 스테레오 카메라 지원 • 소스 디렉터리 구조 구성 	<ul style="list-style-type: none"> • 에러 체크링 코드 대신에 C++ try-catch 문 사용 • OpenMP에서 인텔 TBB(Threading Building Blocks)로 병렬처리 루프 변경 • 윈도우와 Mac OS X에서 64비트 모드에서 OpenCV 빌드 가능 • Mac OS에서 Cocoa와 QTKit 지원
2.2 버전	2.3 버전	2.4 버전
<ul style="list-style-type: none"> • 템플릿 자료구조 추가 • 5개의 기존 라이브러리를 12개의 작은 모듈로 재구성 • 안드로이드 지원 가능 • highgui 모듈에서 16비트 LZW 압축 지원(TIFF 영상) • GPU 처리 지원 	<ul style="list-style-type: none"> • 새롭게 제공되는 바이너리 패키지가 다양한 프리컴파일 라이브러리 포함 • stitching 모듈에서 파노라마 지원 • gpu 모듈에서 CUDA 4.0 지원 	<ul style="list-style-type: none"> • 새 기본 클래스인 cv::Algorithm 도입 • SIFT와 SURF를 유료 모듈로 변경 및 SIFT 성능 대폭 개선 • 캐니 에지 컬러 영상에서 수행
2.4.3 버전	2.4.7 버전	3.0 버전
<ul style="list-style-type: none"> • TBB 설치 없이 기본적인 병렬처리 지원 • OpenCL 컴퓨터 비전 알고리즘인 ocl 모듈 도입 • OpenCV 매니저 개선 • 안드로이드 카메라 지원 개선 	<ul style="list-style-type: none"> • video super-resolution 모듈 도입 • GPU 모듈이 CUDA 5.0까지 지원 • 안드로이드 NDK-r9 지원 • 안드로이드 4.3 지원 	<ul style="list-style-type: none"> • 기존 C++ API 대폭 개선 • cv::Algorithm 적극 사용 • 모바일 CUDA 지원 • IPP, FastCV 같은 저수준 API 지원

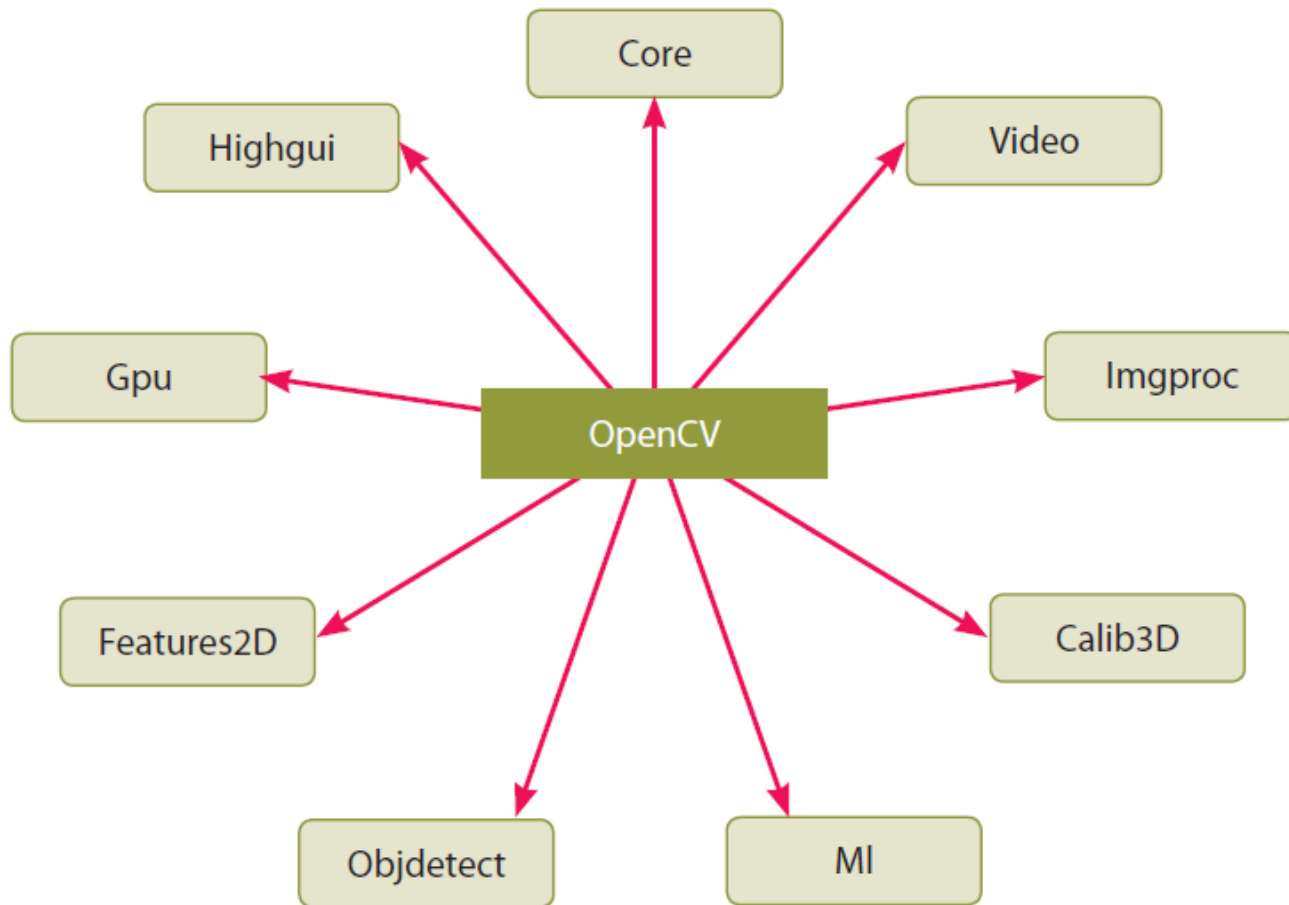
OpenCV의 구성



OpenCV로 할 수 있는 작업

- 영상 파일의 읽기 및 쓰기
- 비디오 캡처 및 저장
- 영상 처리(필터, 변환)
- 영상이나 비디오에서 얼굴, 눈, 자동차와 같은 특정 물체를 감지
- 비디오를 분석하여 움직임을 추정하고, 배경을 없애고, 특정 물체를 추적할 수 있다.
- 기계 학습 알고리즘을 사용하여 물체를 인식할 수 있다.

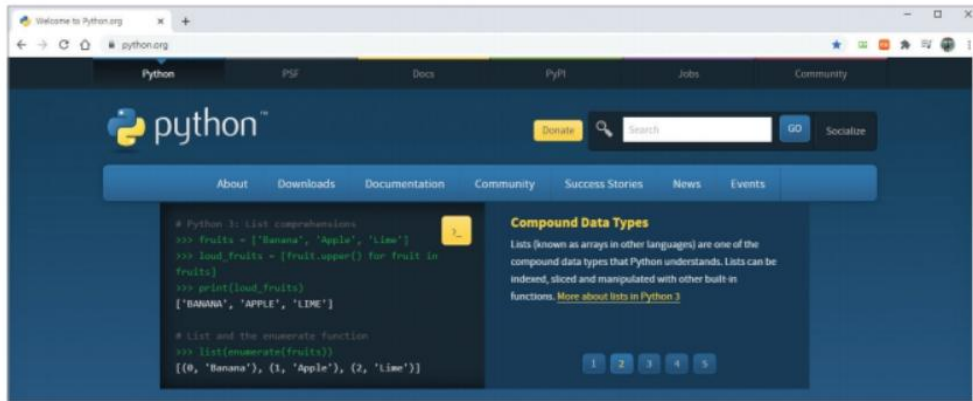
OpenCV 라이브러리 모듈



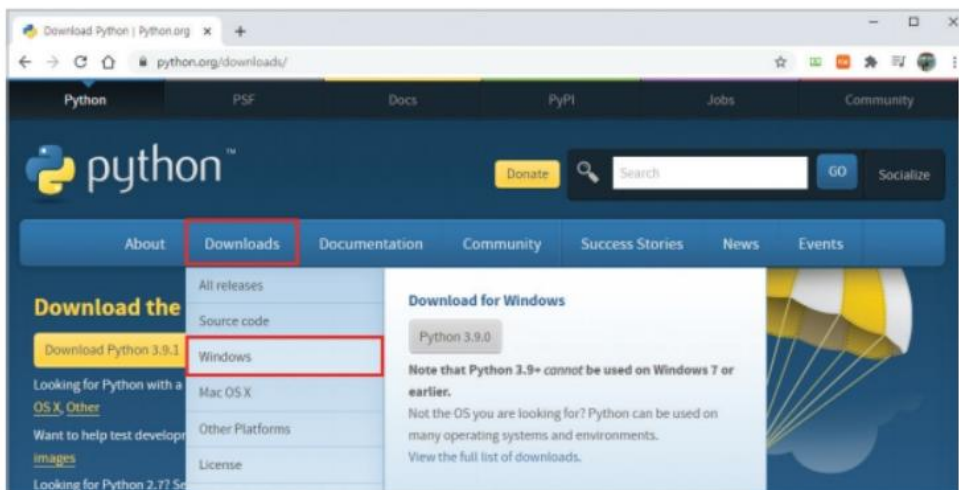
파이썬과 OpenCV 설치

파이썬 설치

- 파이썬 설치
 - 파이썬 공식 홈페이지(<http://python.org>)에 접속



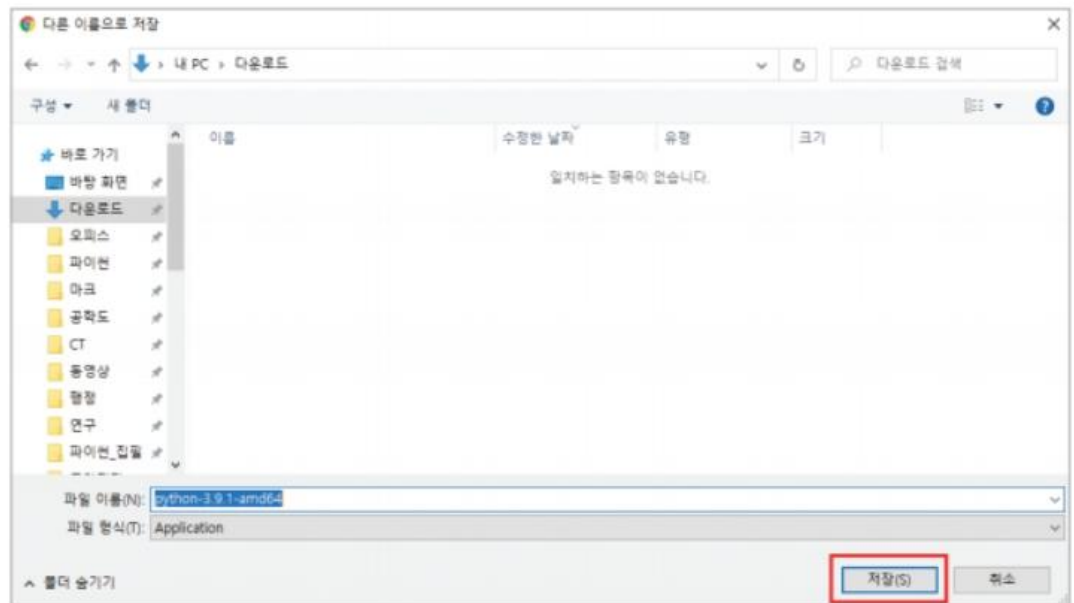
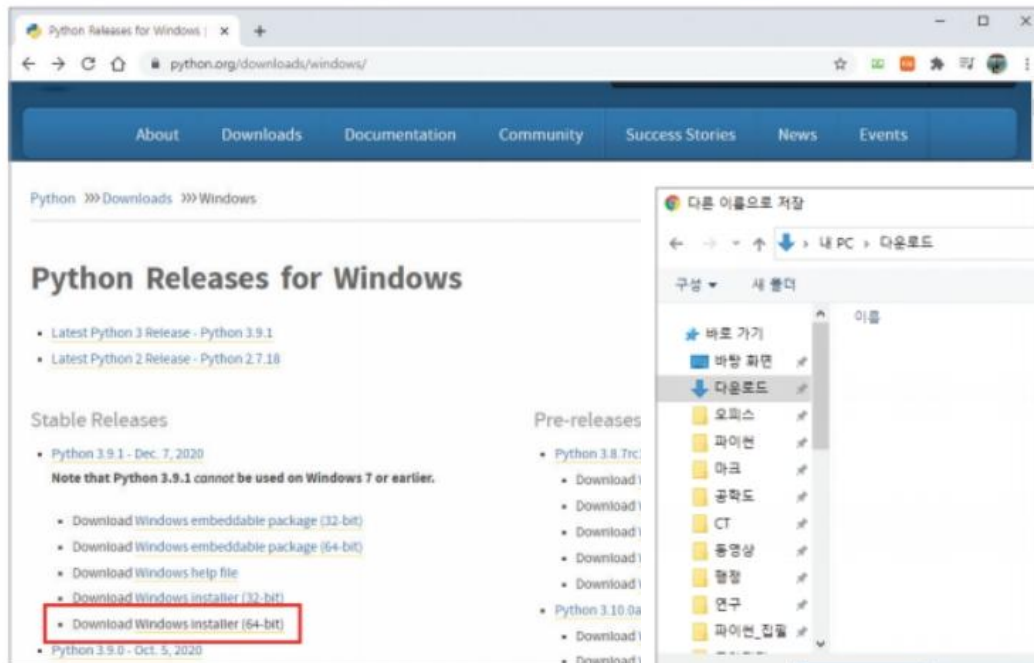
- [Downloads] 메뉴를 클릭하고 [Windows]를 선택



파이썬 설치

- 파이썬 설치

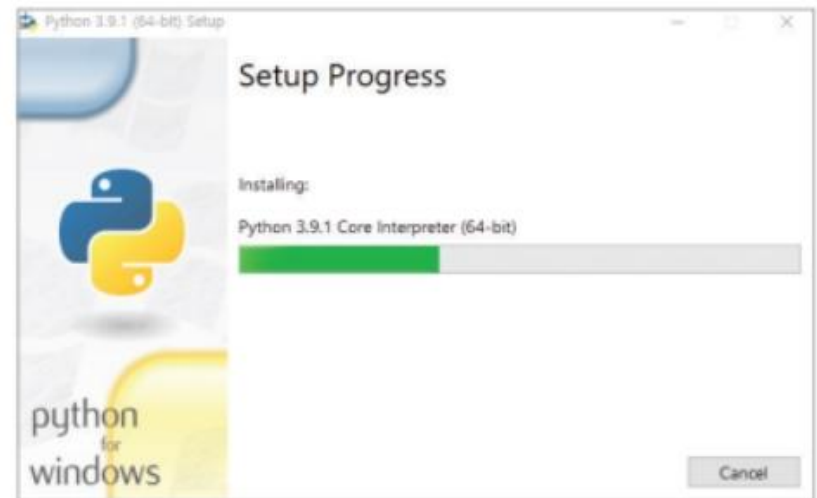
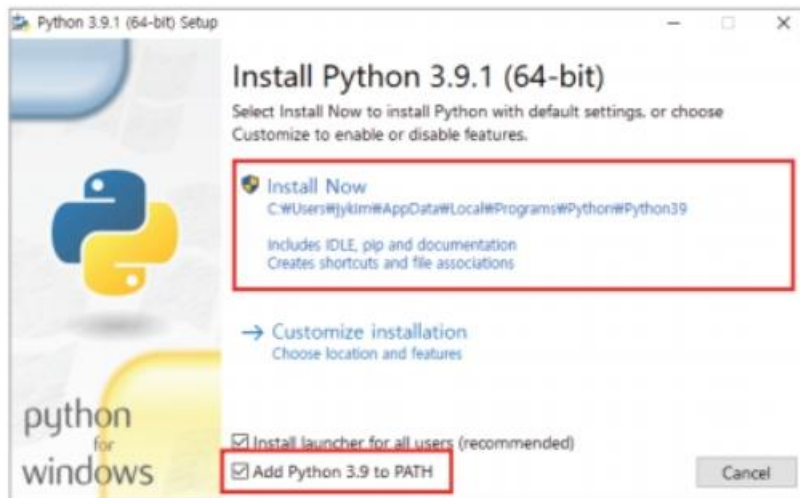
- 최근 버전의 [Download Windows Installer (64-bit)] 클릭
- 설치 파일을 다운로드할 폴더를 선택하고 클릭



파이썬 설치

- 파이썬 설치

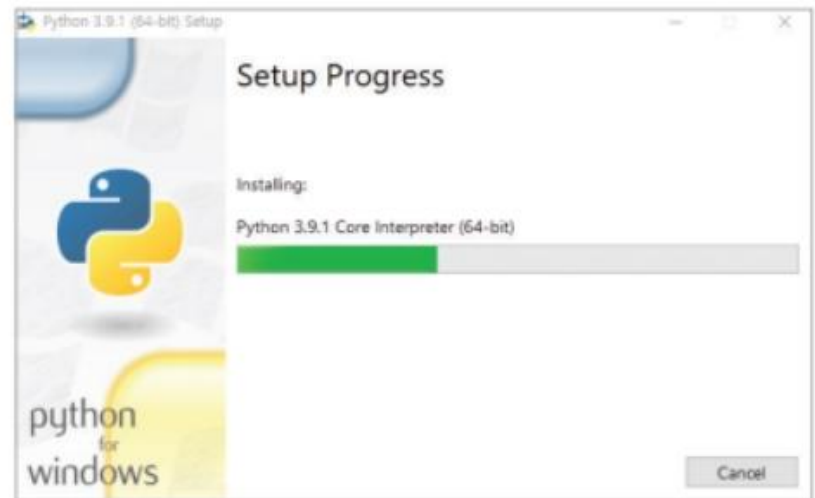
- 다운로드한 폴더에 있는 설치 파일을 실행하고, 설치 시작 화면에서 [Add Python 3.9 to PATH] 항목을 반드시 클릭
- 설치 과정이 끝난 후 설치 성공을 알리는 화면이 나타나면 [Close] 클릭



파이썬 설치

- 파이썬 설치

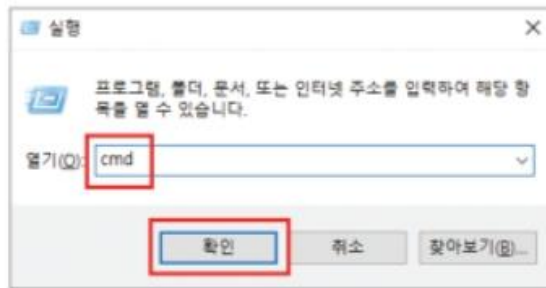
- 다운로드한 폴더에 있는 설치 파일을 실행하고, 설치 시작 화면에서 [Add Python 3.9 to PATH] 항목을 반드시 클릭
- 설치 과정이 끝난 후 설치 성공을 알리는 화면이 나타나면 [Close] 클릭



OpenCV 설치

- OpenCV 설치

- 명령프롬프트 실행 (또는 Windows Power Shell)
- 윈도우 키와 R을 함께 누르면 [실행] 창이 나타나는데, 열기 항목에 'cmd'라고 입력하고 [확인] 클릭



- > pip install opencv-python 명령 실행하여 OpenCV 설치

```
PS C:\Users\naeza> pip install opencv-python
Collecting opencv-python
  Using cached opencv_python-4.11.0.86-cp37-abi3-win_amd64.whl.metadata (20 kB)
Requirement already satisfied: numpy>=1.21.2 in c:\users\naeza\appdata\local\programs\python\python313\lib\site-packages (from opencv-python) (2.2.3)
Using cached opencv_python-4.11.0.86-cp37-abi3-win_amd64.whl (39.5 MB)
Installing collected packages: opencv-python
Successfully installed opencv-python-4.11.0.86
```

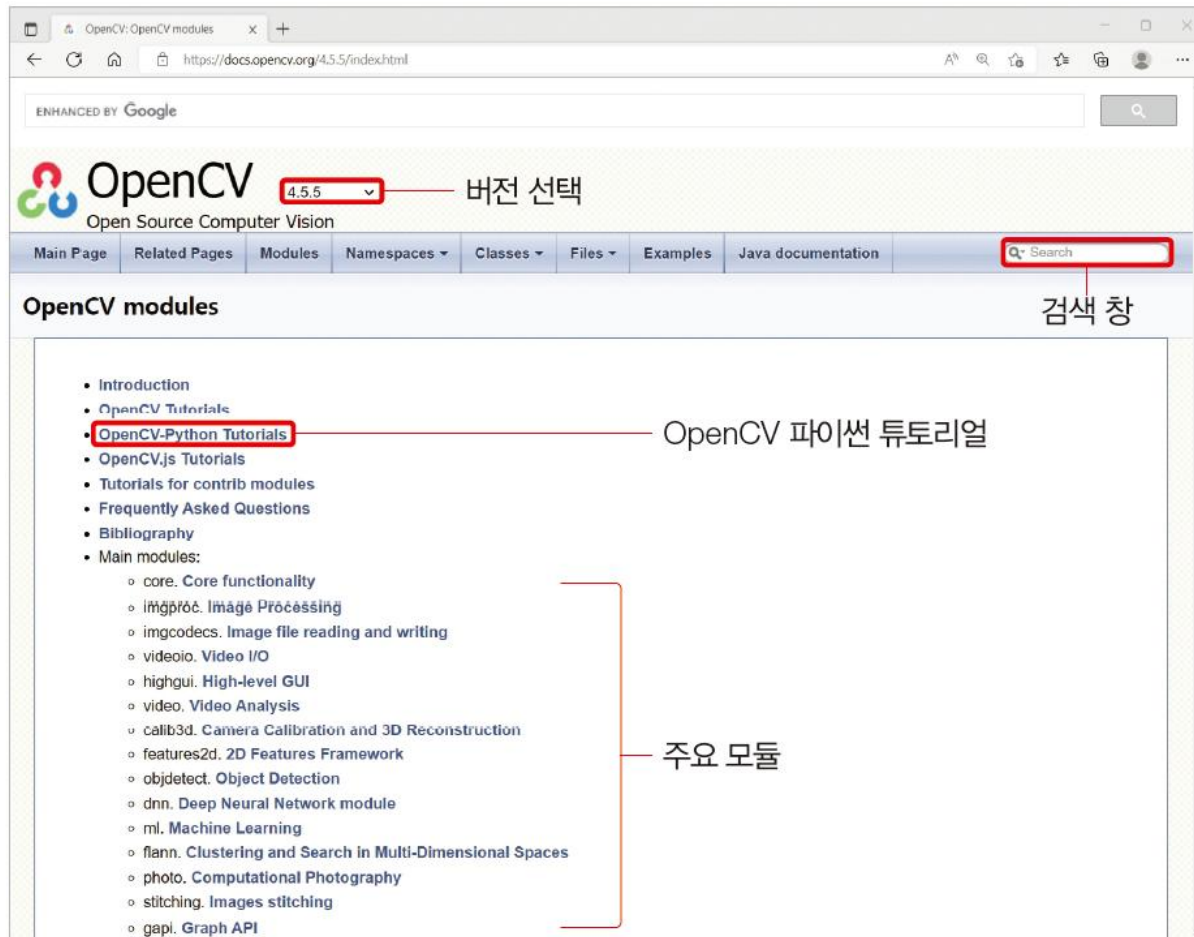
OpenCV 참고 사이트

OpenCV 참고 사이트

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

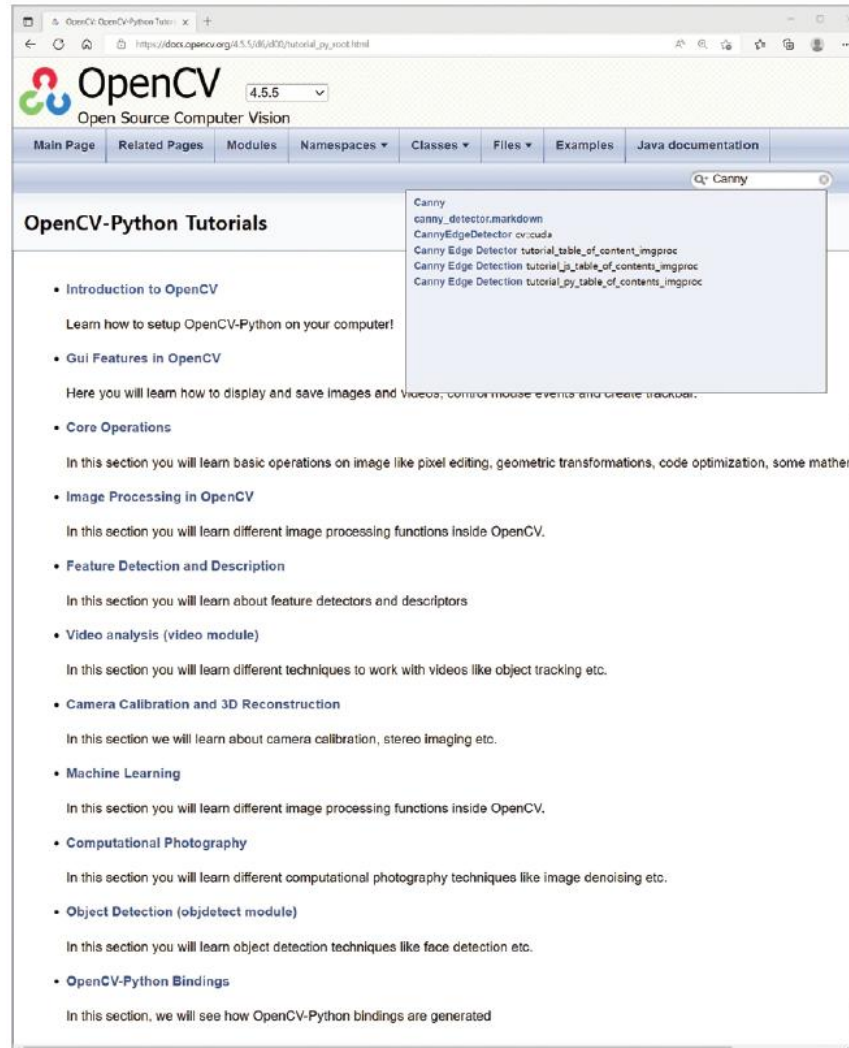
OpenCV 참고 사이트

- OpenCV메뉴얼 사이트
 - <https://docs.opencv.org>



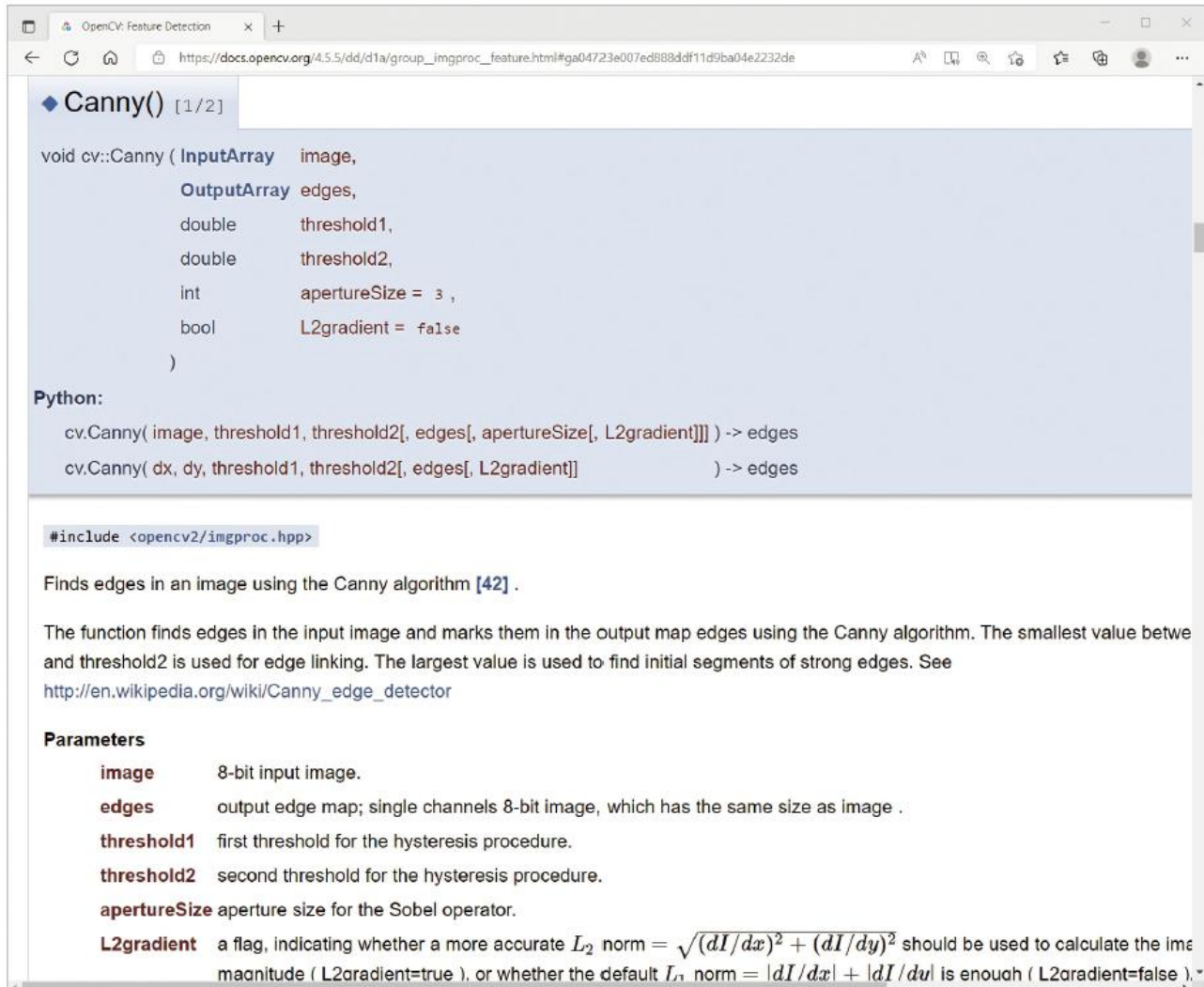
OpenCV 매뉴얼 활용하기

- 유용한 OpenCV-Python 튜토리얼



OpenCV 매뉴얼 활용하기

- 함수를 잘 사용하려면 함수 선언을 잘 살펴야 한다



The screenshot shows the OpenCV documentation for the `Canny()` function. The browser address bar shows the URL: `https://docs.opencv.org/4.5.5/dd/d1a/group_imgproc__feature.html#ga04723e007ed888ddf11d9ba04e2232de`. The function signature is shown in C++ and Python. The C++ signature is: `void cv::Canny (InputArray image, OutputArray edges, double threshold1, double threshold2, int apertureSize = 3, bool L2gradient = false)`. The Python signature is: `cv.Canny(image, threshold1, threshold2[, edges[, apertureSize[, L2gradient]]]) -> edges` and `cv.Canny(dx, dy, threshold1, threshold2[, edges[, L2gradient]]) -> edges`. The documentation includes a header file example: `#include <opencv2/imgproc.hpp>`, a description of the function: "Finds edges in an image using the Canny algorithm [42].", a detailed explanation of the algorithm, a link to the Wikipedia page: `http://en.wikipedia.org/wiki/Canny_edge_detector`, and a list of 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), and `L2gradient` (a flag, indicating whether a more accurate L_2 norm should be used to calculate the image magnitude (`L2gradient=true`), or whether the default L_1 norm is enough (`L2gradient=false`).

객체지향의 이해

객체 다루기

- 파이썬은 객체지향 언어
 - 객체지향은 컴퓨터 비전 프로그래밍에 매우 유리

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

객체지향 잘 활용하기

- 객체의 클래스를 알려주는 `type`과 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.
```

객체 확인하기

- 객체의 클래스를 알려주는 `type`과 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.
```

OpenCV 사용하기

OpenCV로 이미지 읽고 표시하기

- Python 소스코드가 있는 디렉터리에 파일을 다운로드 (soccer.jpg)
 - 본인이 원하는 이미지를 인터넷에서 다운로드 해서 응용 가능

```
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)   # 윈도우에 영상 표시
10
11 cv.waitKey()
12 cv.destroyAllWindows()
```



OpenCV로 이미지 읽고 표시하기

- OpenCV 영상은 `numpy.ndarray` 클래스 형의 객체
- `numpy`는 다차원 배열을 위한 사실상 표준 모듈
 - OpenCV는 영상을 `numpy.ndarray`로 표현
 - OpenCV는 `numpy`의 다양한 함수 사용 가능

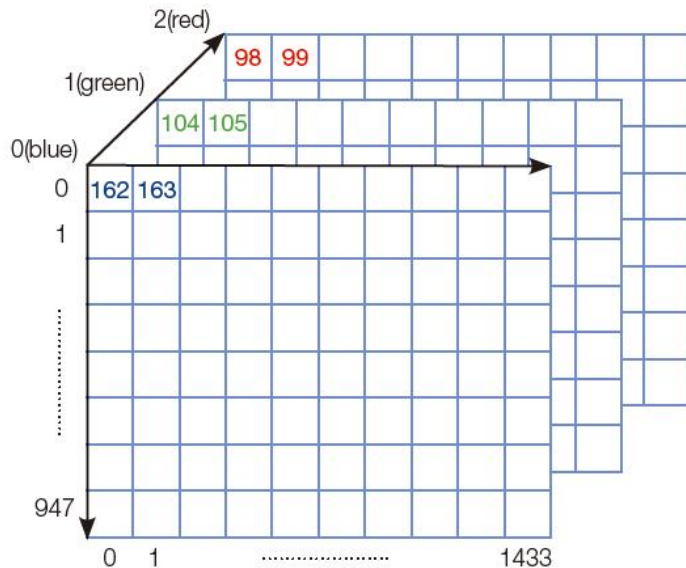
```
In [1]: type(img)
        numpy.ndarray
In [2]: img.shape
        (948,1434,3)
```


OpenCV로 이미지 읽고 표시하기

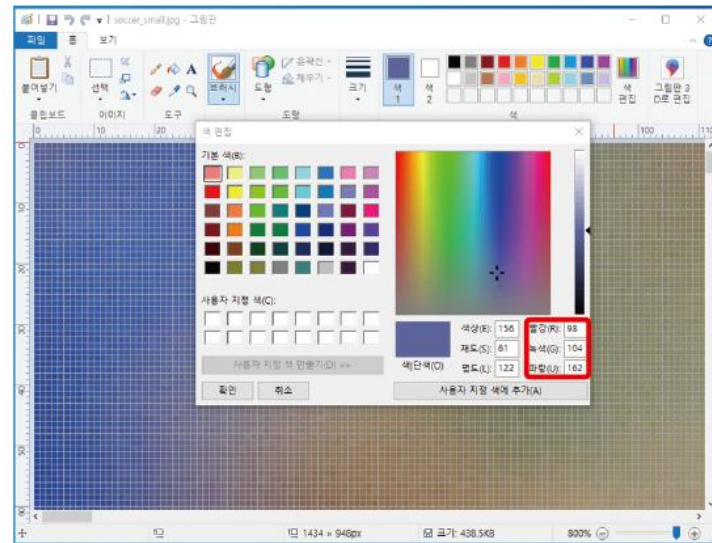
- 영상의 표현
 - 화소의 위치 (r,c) 또는 (y,x)
 - 화소값 조사

```
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

OpenCV로 이미지 변환하기

- 영상 형태 변환하고 크기 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,dsi=(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()
```

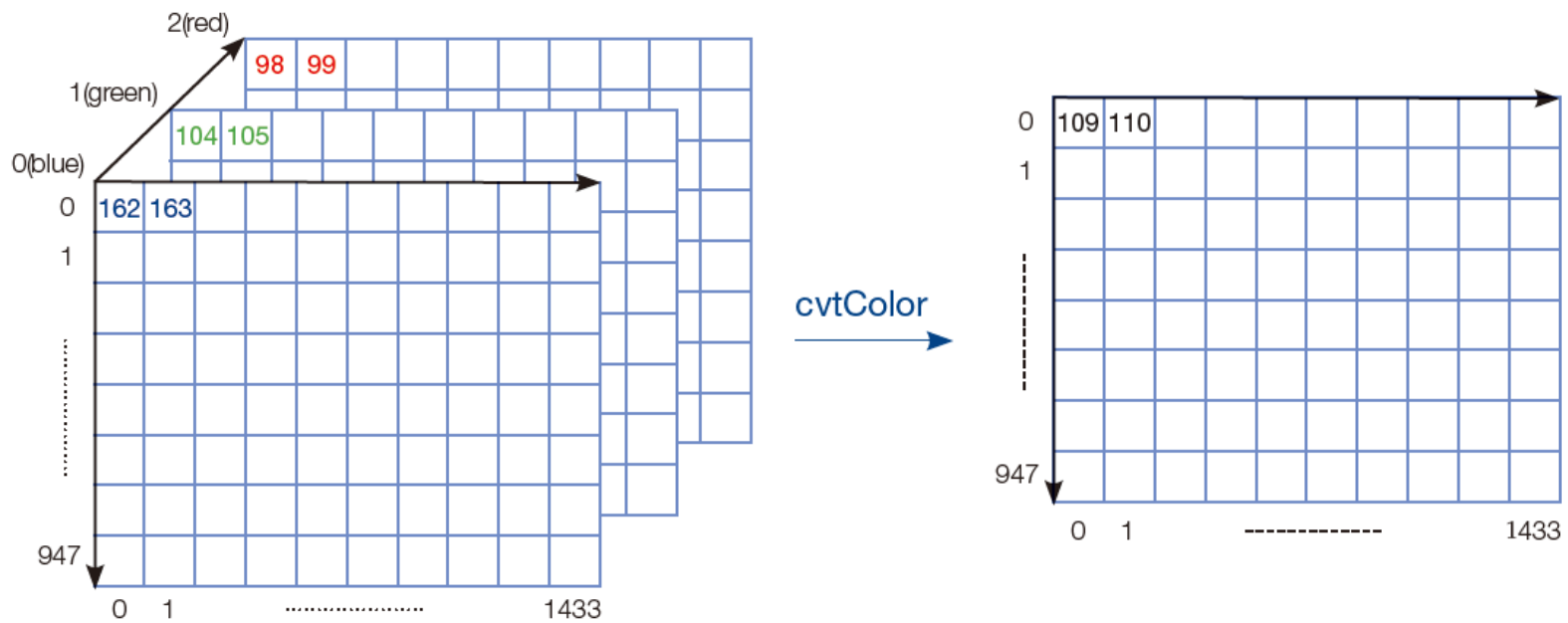


OpenCV로 이미지 읽고 표시하기

- cvtColor 함수가 컬러 영상을 명암 영상으로 바꾸는 방법

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

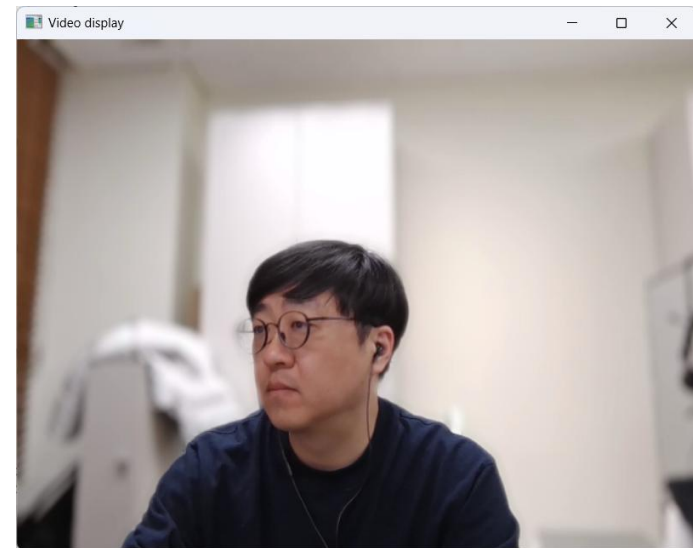
(참조: https://docs.opencv.org/3.4/de/d25/imgproc_color_conversions.html)



OpenCV로 웹캠 활용하기

- 웹 캠에서 비디오 읽기

```
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()
```



OpenCV로 웹캠 활용하기

- 비디오에서 영상 수집하기(numpy의 hstack 함수 사용)

```
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
```

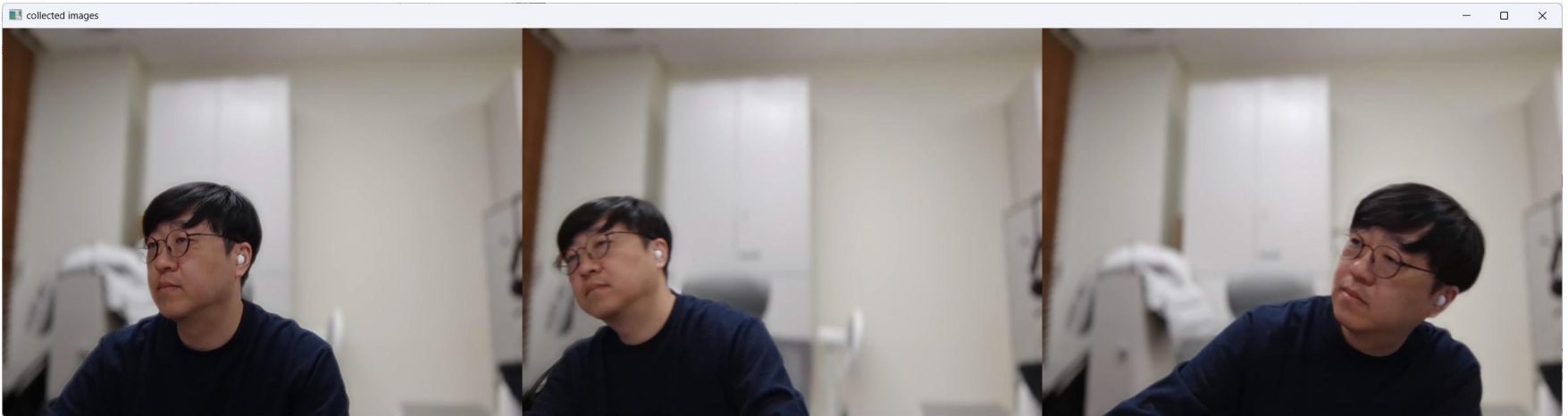
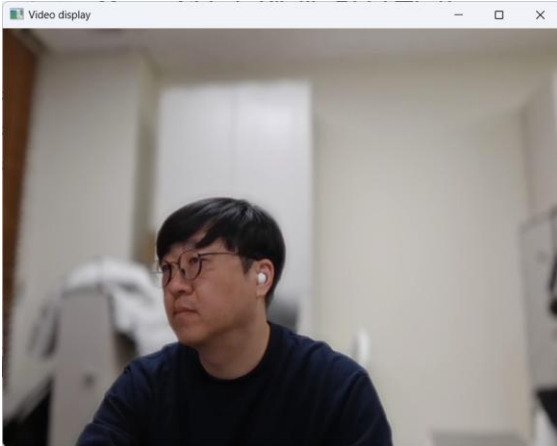
OpenCV로 웹캠 활용하기

- 비디오에서 영상 수집하기(numpy의 hstack 함수 사용)

```
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()
```

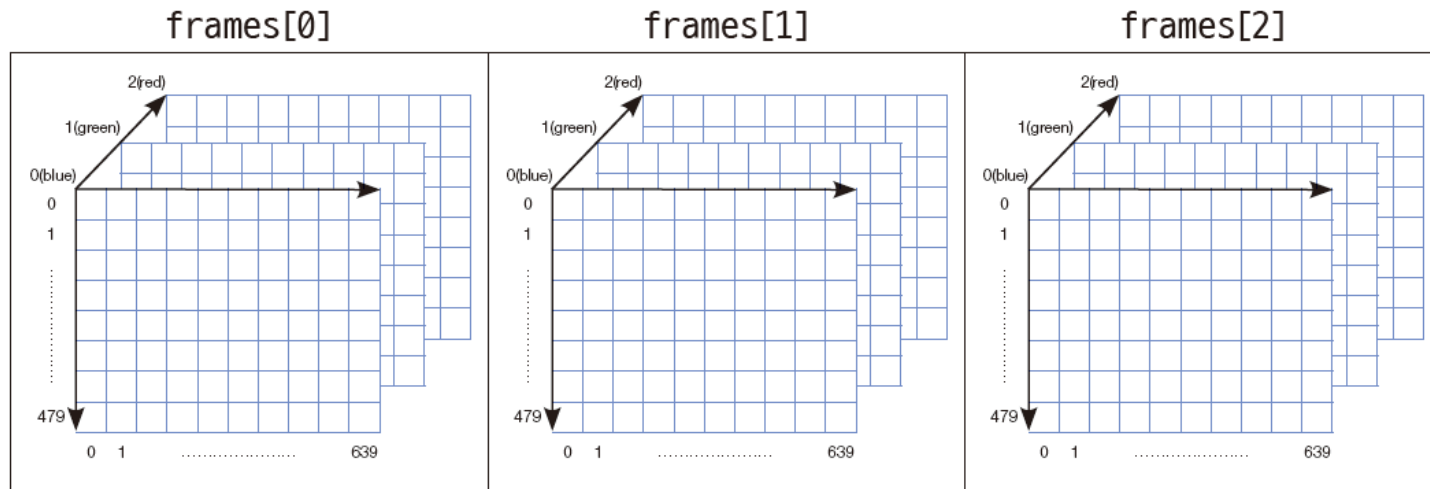
OpenCV로 웹캠 활용하기

- 비디오에서 영상 수집하기(numpy의 hstack 함수 사용)
 - 웹캠 영상이 출력될 때 **c키**를 누름
 - **q키**를 눌러서 종료하면 스택에 추가되었던 frame이 표시됨

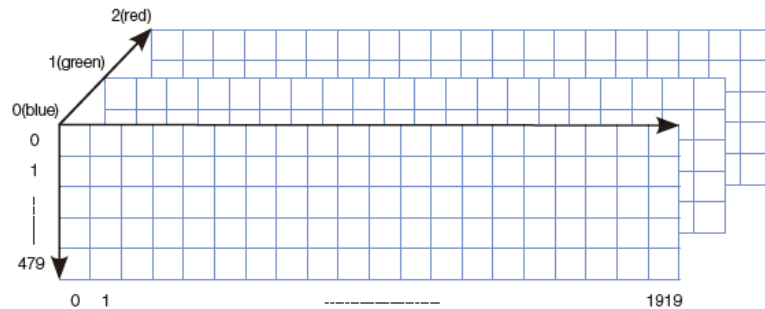


OpenCV로 웹캠 활용하기

- 저장된 frame의 자료구조



(a) frames 리스트



(b) imgs 배열

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

그래픽 기능과 사용자 인터페이스 만들기

- 영상에 도형을 그리고 글씨 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

그래픽 기능과 사용자 인터페이스 만들기

- 영상에 도형을 그리고 글씨 쓰기



그래픽 기능과 사용자 인터페이스 만들기

- 함수 선언에 대한 이해
 - OpenCV 공식사이트의 `rectangle()` 함수의 선언

◆ `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 <code>pt1</code> .
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.

그래픽 기능과 사용자 인터페이스 만들기

- 마우스를 통한 상호작용(콜백 함수에 대한 이해 필요)

```
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
```


그래픽 기능과 사용자 인터페이스 만들기

- 마우스를 통한 상호작용



그래픽 기능과 사용자 인터페이스 만들기

- 마우스 드래그로 도형 크기 조절하기

```
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 직사각형 그리기
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
```


그래픽 기능과 사용자 인터페이스 만들기

- 마우스를 통한 상호작용



페인팅 기능

- 페인팅 기능 사용하기

```
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
```

페인팅 기능

- 페인팅 기능 사용하기

```
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
```

페인팅 기능

- 페인팅 기능 사용하기

