

패턴 인식

OPENCV 시작

설치하기

- 교재는 아나콘다를 이용
- 우리는 파이참+OPENCV를 사용 예정

OpenCV 매뉴얼 활용하기

■ OpenCV-Python 튜토리얼을 잘 살필 것

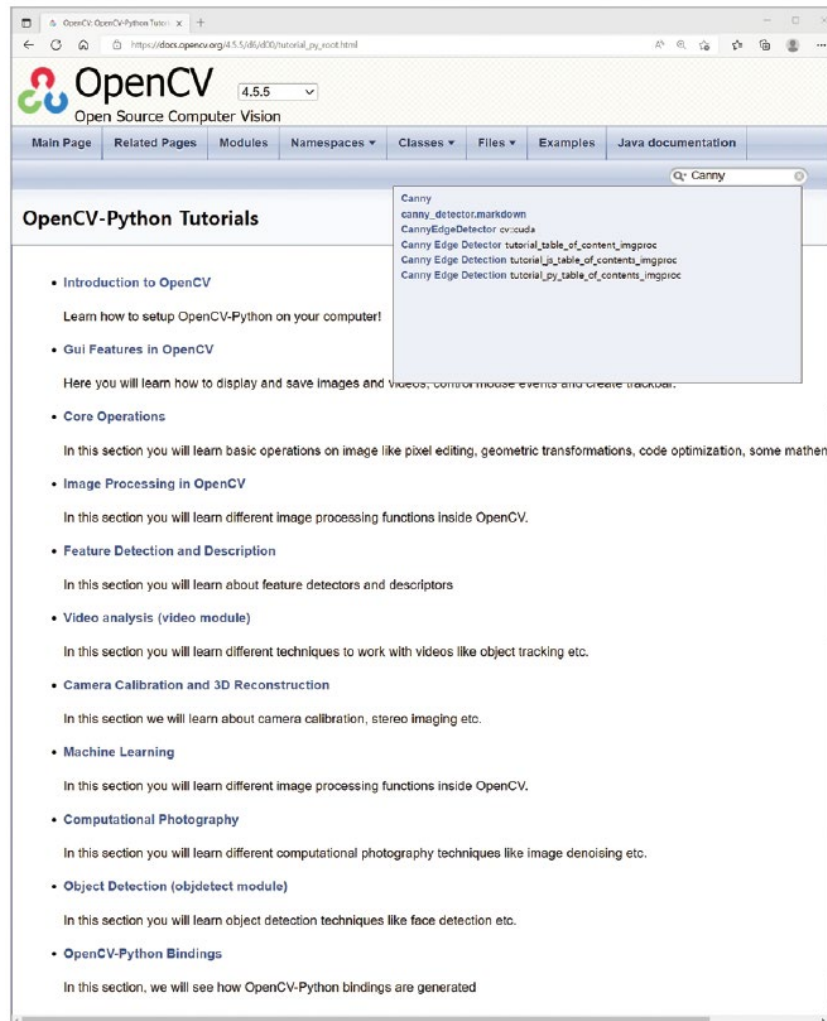
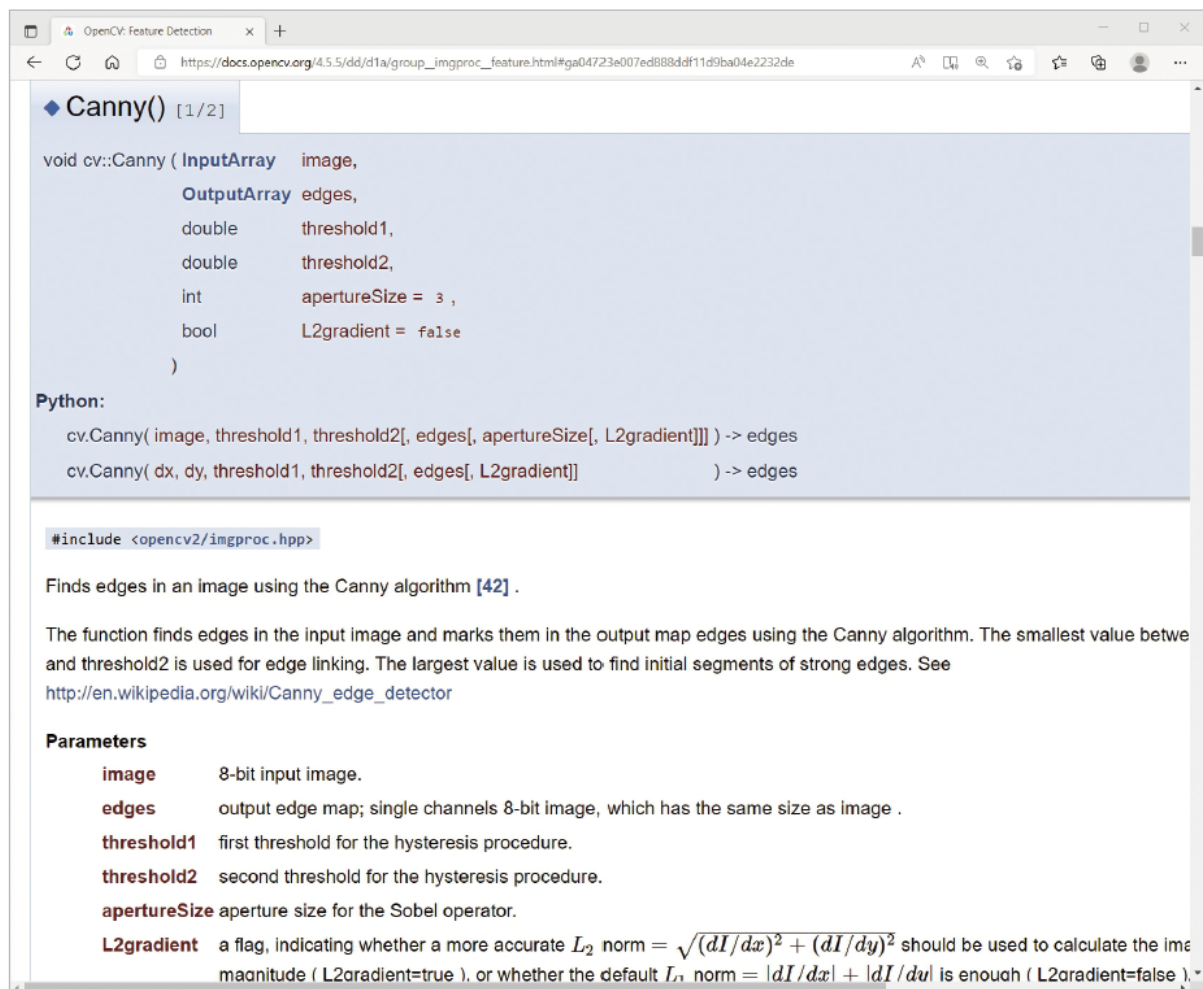


그림 2-6 OpenCV-Python 튜토리얼 화면

OpenCV 매뉴얼 활용하기

■ 함수 선언을 잘 살필 것



◆ Canny() [1/2]

```
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

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 gradient 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 장점

- 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

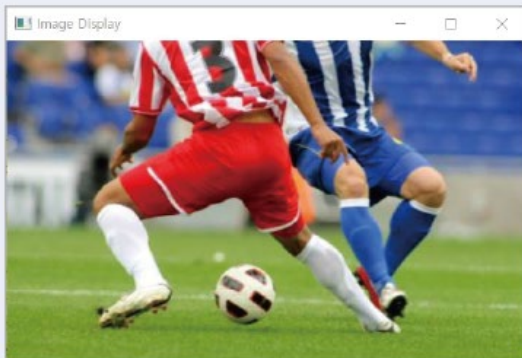
2.4 [프로그래밍 예제1] 영상을 읽고 표시하기

■ 처음 해보는 OpenCV 프로그래밍

프로그램 2-2

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



OpenCV에서 영상은 `numpy.ndarray` 클래스 형의 객체

■ `numpy`는 다차원 배열을 위한 사실상 표준 모듈

- 이런 이유로 OpenCV는 영상을 `numpy.ndarray`로 표현
- OpenCV가 다루는 영상은 `numpy`가 제공하는 다양한 기능(함수)을 사용할 수 있음

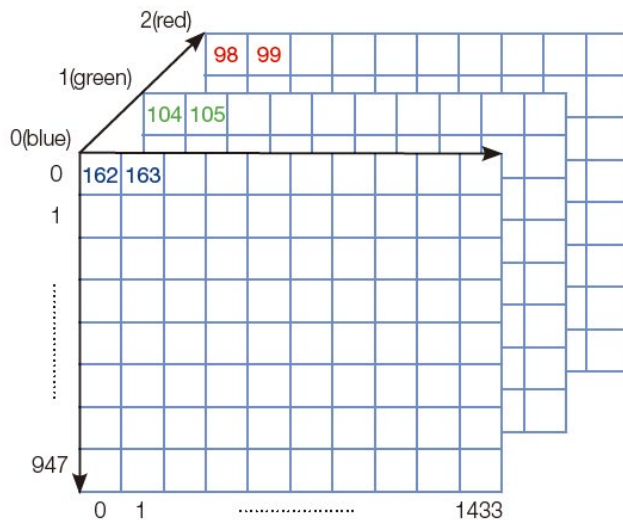
```
In [1]: type(img)
        numpy.ndarray
In [2]: img.shape
        (948,1434,3)
```

OpenCV에서 영상은 `numpy.ndarray` 클래스 형의 객체

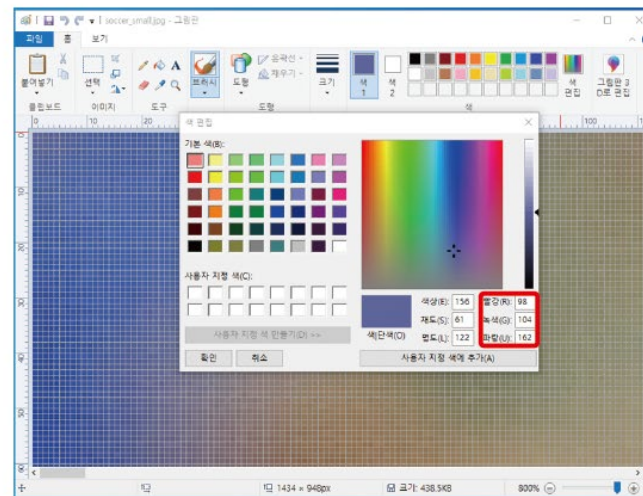
■ 영상의 표현

- 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

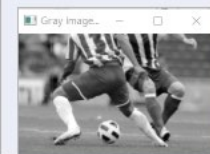
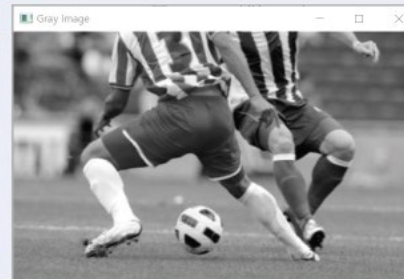
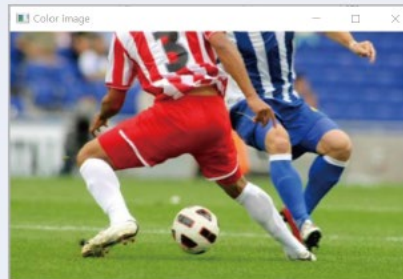
2.5 [프로그래밍 예제2] 영상 형태 변환하고 크기 축소하기

■ [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,dsiz=(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)

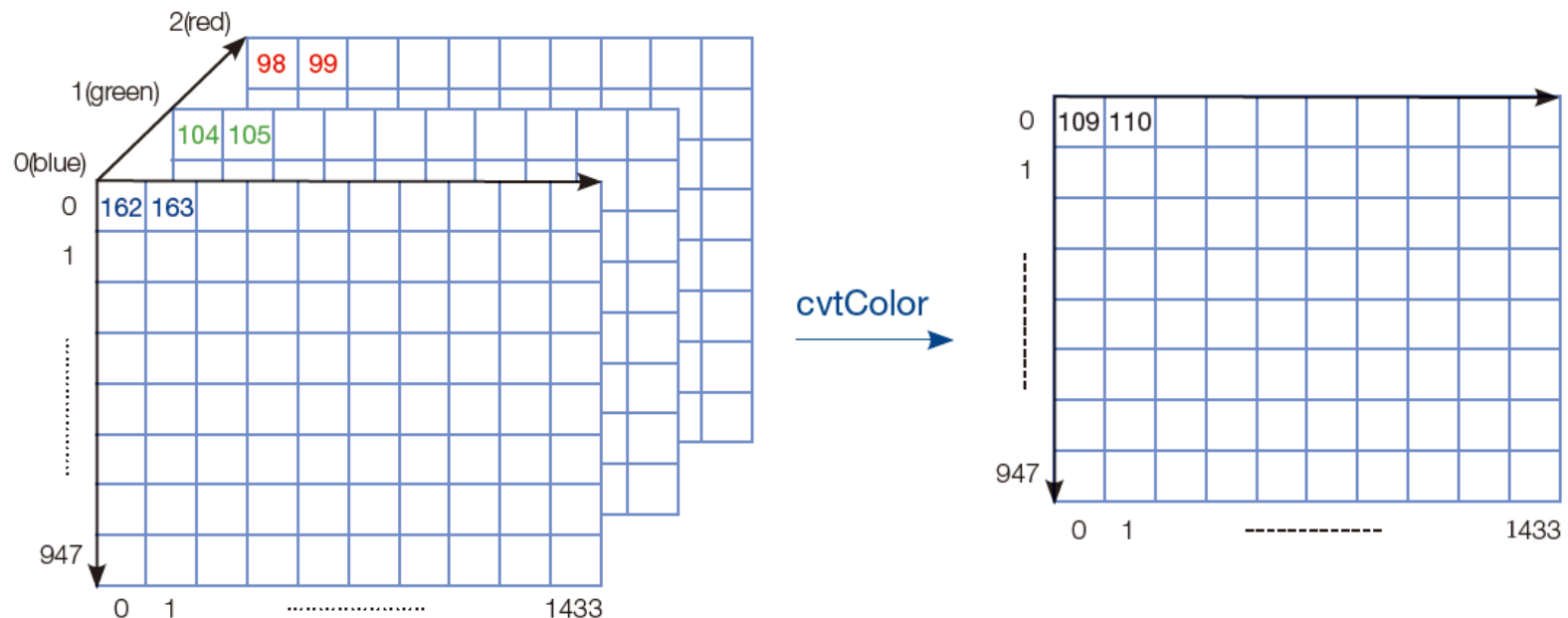


그림 2-10 BGR 컬러 영상을 명암 영상으로 변환

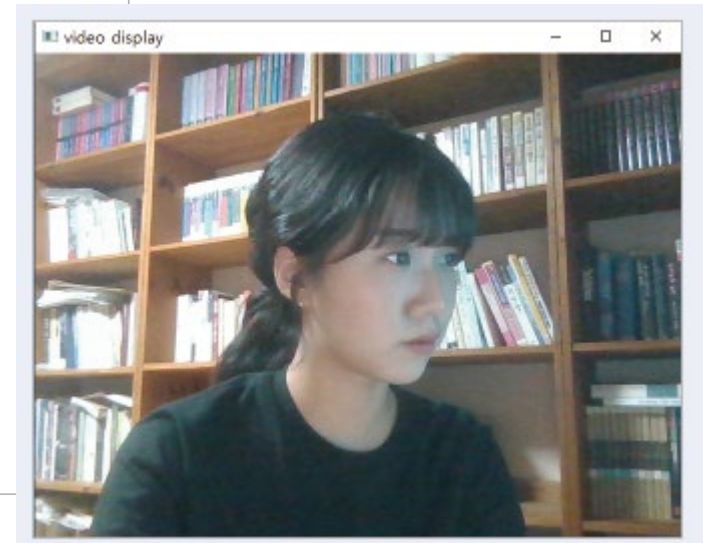
2.6 [프로그래밍 예제3] 웹 캠에서 비디오 읽기

■ 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

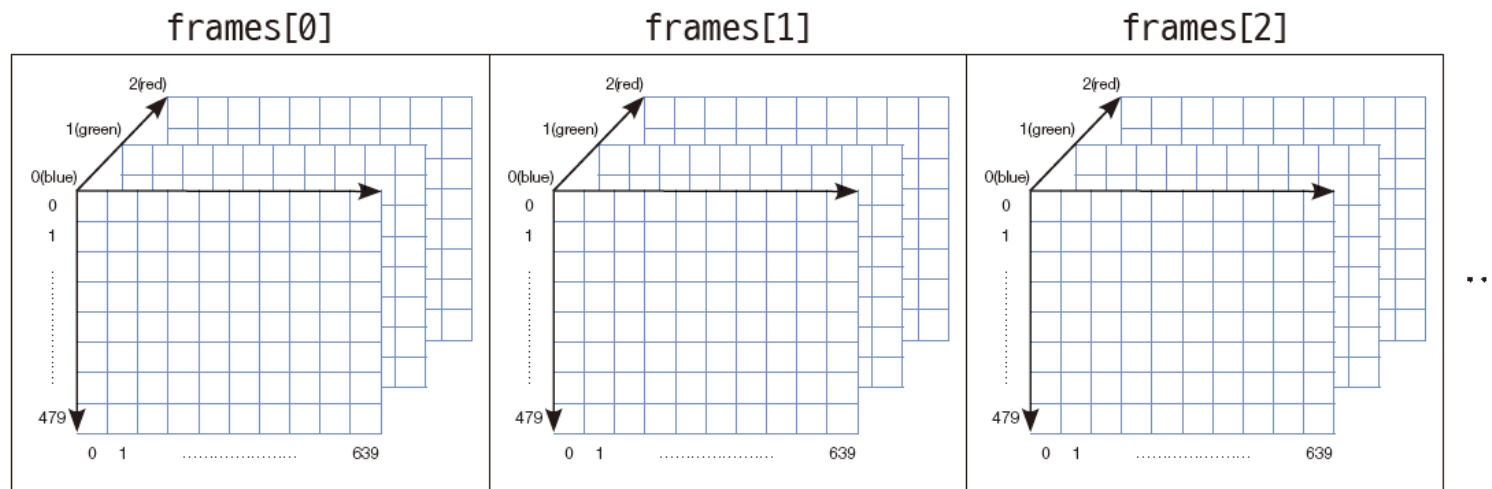
```
18     cv.imshow('Video display',frame)
19
20     key=cv.waitKey(1)                # 1밀리초 동안 키보드 입력 기다림
21     if key==ord('c'):                # 'c' 키가 들어오면 프레임을 리스트에 추가
22         frames.append(frame)
23     elif key==ord('q'):               # 'q' 키가 들어오면 루프를 빠져나감
24         break
25
26 cap.release()                       # 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

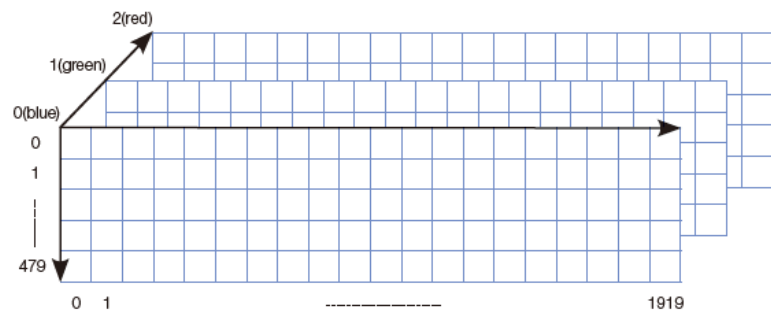


2.6 [프로그래밍 예제3] 웹 캠에서 비디오 읽기

■ [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()
```



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 <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.

그림 2-12 OpenCV 공식 사이트가 제공하는 `rectangle` 함수의 선언(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 클릭했
11         cv.rectangle(img,(x,y),(x+200,y+200),(0,0,255),2)
12     elif event==cv.EVENT_RBUTTONDOWN: # 마우스 오른쪽 버튼 클릭
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
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

