

# OpenCV Practice

---

Computer Vision and Pattern Recognition

Byeongjoon Noh

powernoh@sch.ac.kr



# 예제 1 – 이미지를 읽고 표시하기

- 이미지 파일을 읽고 윈도우에 디스플레이 하기

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()
```



- 참고

- OpenCV에서 이미지(영상)은 `numpy.ndarray` 클래스의 객체
- OpenCV 함수의 대부분은 이미지를 `numpy` 형태로 입력받음

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

## 예제 2 – 이미지 저장하기

---

- 이미지 파일을 읽고 윈도우에 디스플레이 하기

```
import cv2
import sys

# 이미지 읽기
img = cv2.imread('../data/apple.jpg')

if img is None:
    sys.exit('파일을 찾을 수 없습니다')

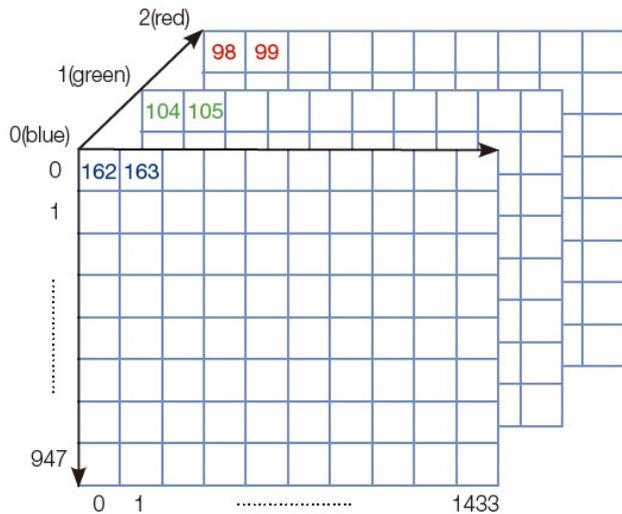
# 이미지 보기
cv2.imshow('Image Window', img)
cv2.waitKey(0) # 키보드 입력을 기다린다.
cv2.destroyAllWindows()

cv2.imwrite('../data/apple_copy.jpg', img)
```

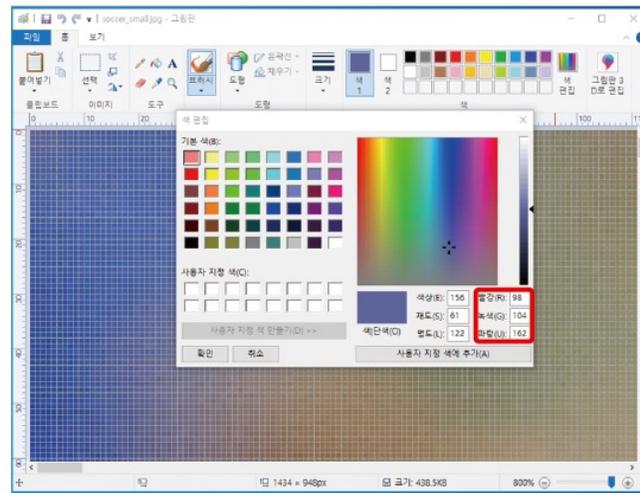
# 영상의 표현

- 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

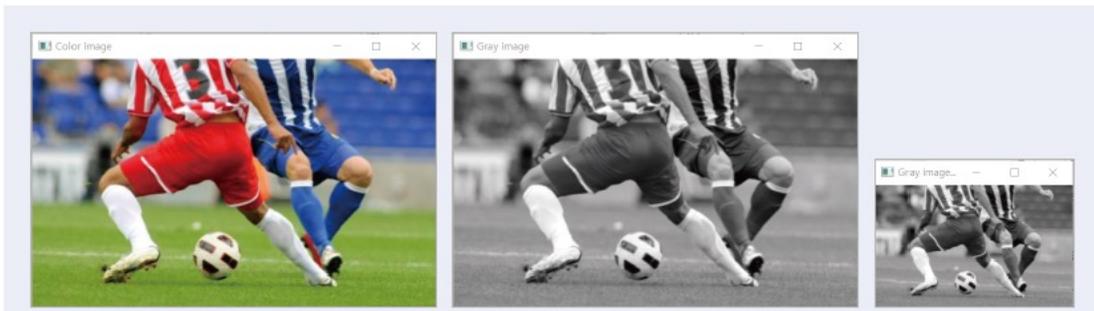
그림 2-9 img 객체가 표현하는 영상의 구조와 내용

## 예제 2 – 영상의 형태를 변환하고 크기 축소하기

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()
```



## 예제 2 – 영상의 형태를 변환하고 크기 축소하기

- cvtColor 함수가 컬러 영상을 명암 영상으로 바꾸는 방법

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

- 참고: [https://docs.opencv.org/4.7.0/de/d25/imgproc\\_color\\_conversions.html](https://docs.opencv.org/4.7.0/de/d25/imgproc_color_conversions.html)

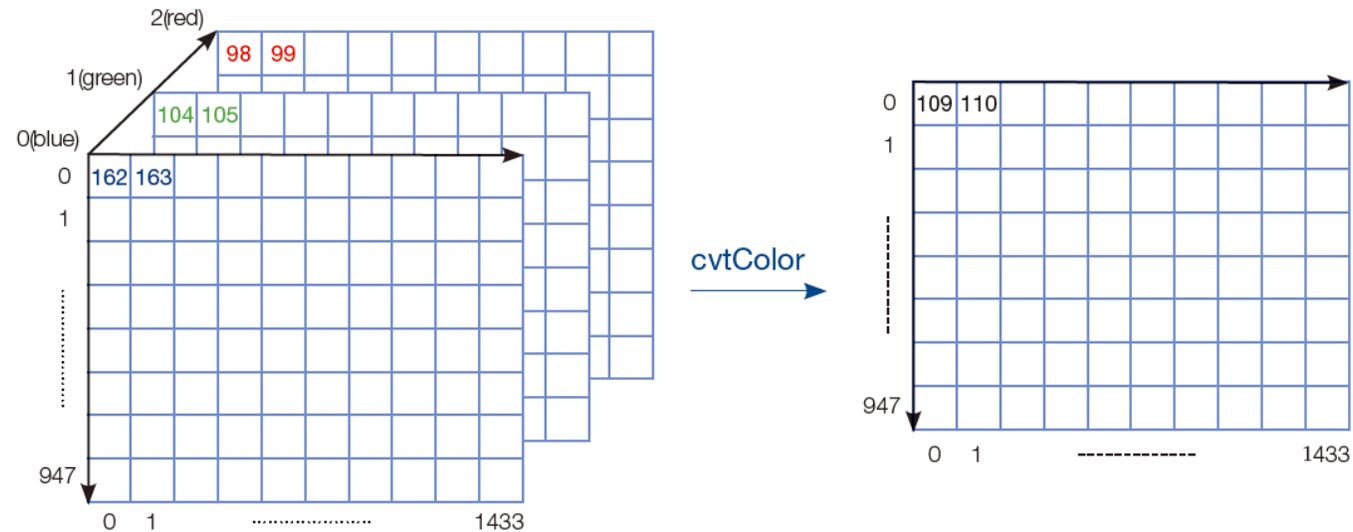


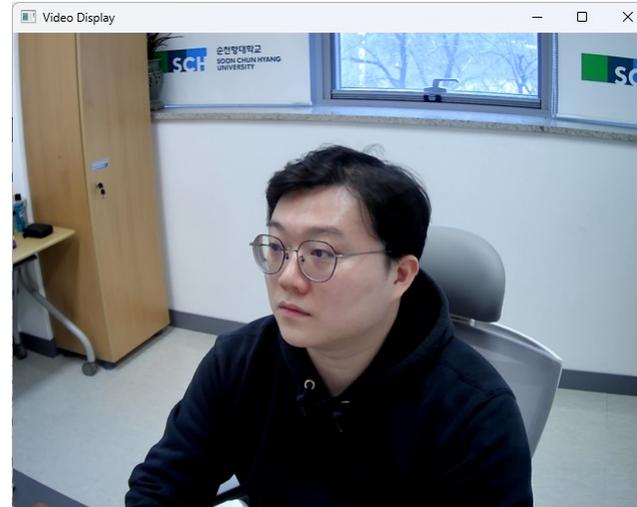
그림 2-10 BGR 컬러 영상을 명암 영상으로 변환

## 예제 3 – 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()
```



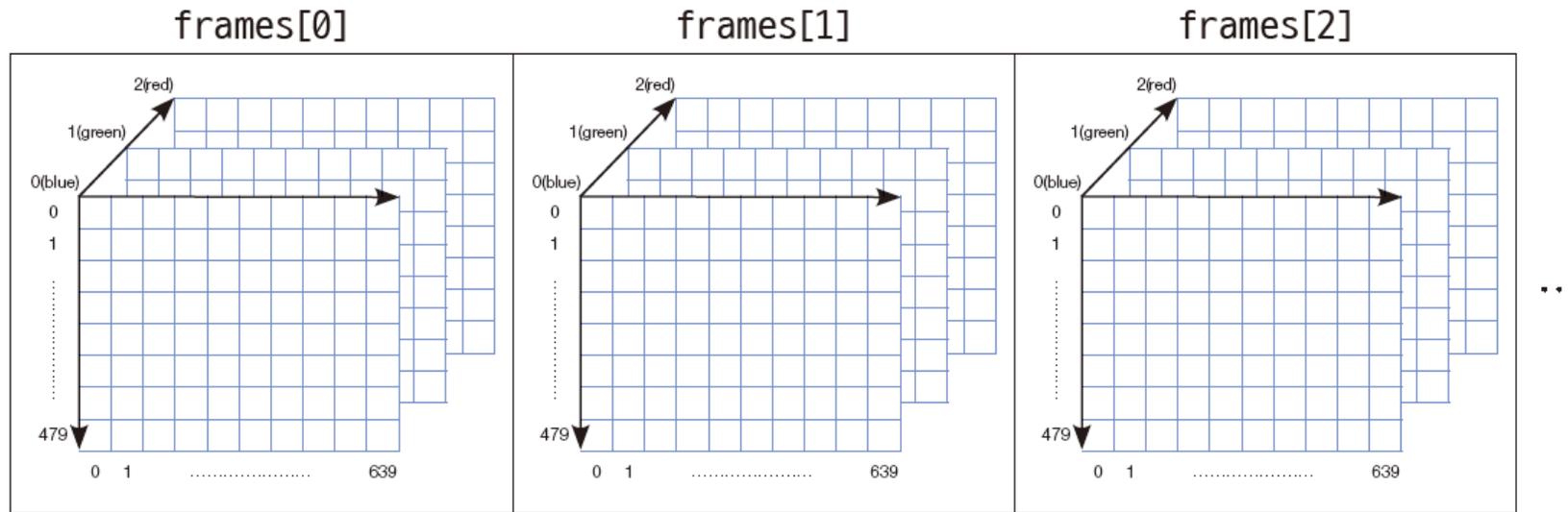
## 예제 4 – 웹 캠에서 비디오 읽기 + 영상 수집 하기

- numpy의 hstack 함수 사용

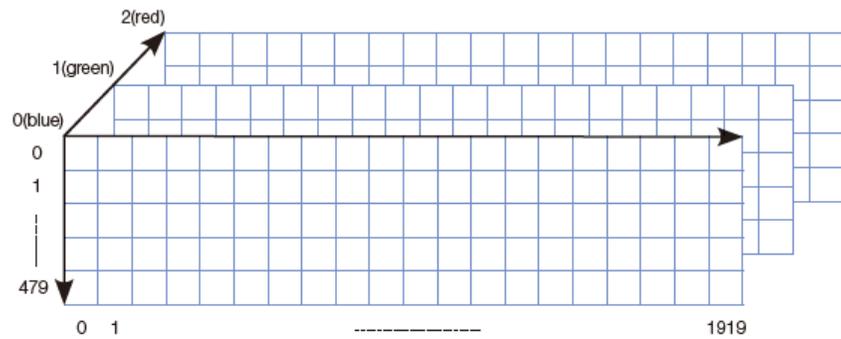
프로그램 2-5 비디오에서 수집한 영상을 이어 붙이기

```
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
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()
```

# 비디오 수집 시 자료구조



(a) frames 리스트



(b) imgs 배열

그림 2-11 [프로그램 2-5]의 자료 구조

## 예제 5 – 그래픽 기능과 사용자 인터페이스

---

- OpenCV의 그래픽 기능
  - 영상에 글씨나 도형을 넣는데 유용함
  - line, rectangle, polylines, circle, ellipse, putText 함수 등

## 예제 5 – 그래픽 기능과 사용자 인터페이스

- 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



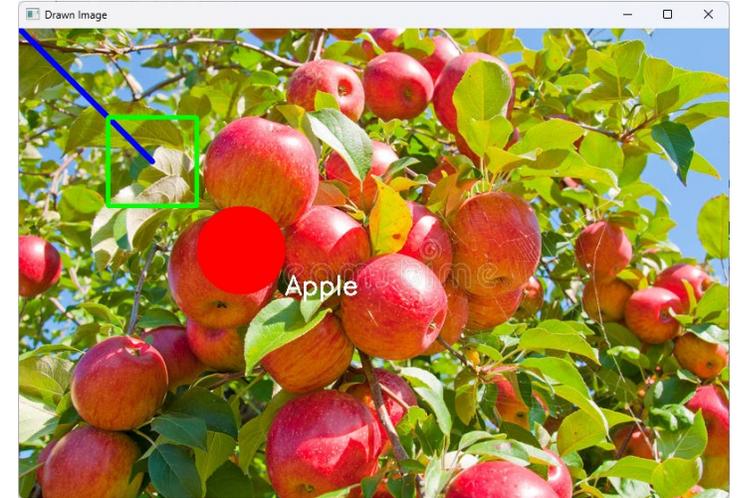
## 예제 5 – 그래픽 기능과 사용자 인터페이스

- `cv2.line()`: 이미지에 선을 그림
  - `img`: 선을 그릴 이미지
  - `pt1`: 선의 시작점 좌표 (`x, y`)
  - `pt2`: 선의 끝점 좌표 (`x, y`)
  - `color`: 선의 색상. BGR 형식의 튜플 (Blue, Green, Red)
  - `thickness`: 선의 두께. 픽셀 단위
  - `lineType`: 선의 유형. `cv2.LINE_8`, `cv2.LINE_AA` 등
  - `shift`: 좌표의 소수점 이하 비트 수. 좌표의 정밀도를 조정
  
- 예시: `cv2.line(img, (0, 0), (150, 150), (255, 0, 0), 5) # 파란색 선`



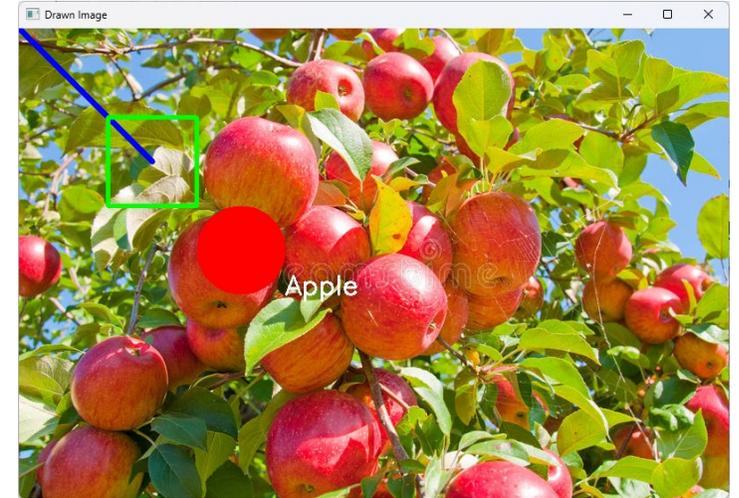
## 예제 5 – 그래픽 기능과 사용자 인터페이스

- `cv2.rectangle()`: 이미지에 사각형을 그림
  - `img`: 사각형을 그릴 이미지
  - `pt1`: 사각형의 왼쪽 상단 모서리 좌표 (`x, y`)
  - `pt2`: 사각형의 오른쪽 하단 모서리 좌표 (`x, y`)
  - `color`: 사각형의 색상. BGR 형식의 튜플
  - `thickness`: 테두리의 두께. 픽셀 단위. `-1` = 사각형 내부를 채움
  - `lineType`: 선의 유형
  - `shift`: 좌표의 소수점 이하 비트 수.
  
- 예시: `cv2.rectangle(img, (100, 100), (200, 200), (0, 255, 0), 3) # 초록색 사각형`



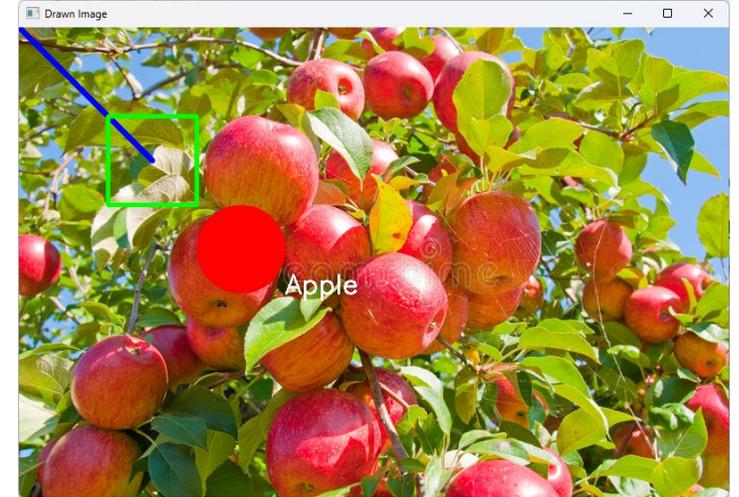
## 예제 5 – 그래픽 기능과 사용자 인터페이스

- `cv2.circle()`: 이미지에 원을 그림
  - `img`: 원을 그릴 이미지
  - `center`: 원의 중심 좌표 (`x`, `y`)
  - `radius`: 원의 반지름. 픽셀 단위
  - `color`: 원의 색상. BGR 형식의 튜플
  - `thickness`: 테두리의 두께. 픽셀 단위. `-1` = 원 내부를 채움
  - `lineType`: 선의 유형
  - `shift`: 좌표의 소수점 이하 비트 수
  
- 예시: `cv2.circle(img, (250, 250), 50, (0, 0, 255), -1)` # 빨간색 원



## 예제 5 – 그래픽 기능과 사용자 인터페이스

- `cv2.putText()`: 이미지에 텍스트를 씌움
  - `img`: 텍스트를 씌울 이미지
  - `text`: 쓰고자 하는 텍스트
  - `org`: 텍스트가 시작되는 위치의 좌표 (`x, y`)
  - `fontFace`: 폰트 유형. 예: `cv2.FONT_HERSHEY_SIMPLEX`
  - `fontScale`: 폰트 크기의 배율 인자
  - `color`: 텍스트의 색상. BGR 형식의 튜플
  - `thickness`: 텍스트 선의 두께. 픽셀 단위
  - `lineType`: 선의 유형
  - `bottomLeftOrigin`: True일 경우, 이미지의 좌측 하단을 원점으로 사용
- 예시: `cv2.putText(img, 'Apple', (300, 300), cv2.FONT_HERSHEY_SIMPLEX, 1, (255, 255, 255), 2)`



# 예제 5 – 그래픽 기능과 사용자 인터페이스

- 다양한 OpenCV 함수에 대한 document

- [https://docs.opencv.org/4.7.0/d6/d6e/group\\_imgproc\\_draw.html#gaf10604b069374903dbd0f0488cb43670](https://docs.opencv.org/4.7.0/d6/d6e/group_imgproc_draw.html#gaf10604b069374903dbd0f0488cb43670)

◆ circle()

```
void cv::circle ( InputOutputArray img,
                 Point center,
                 int radius,
                 const Scalar & color,
                 int thickness = 1 ,
                 int lineType = LINE_8 ,
                 int shift = 0
                 )
```

Python:

```
cv.circle( img, center, radius, color[, thickness[, lineType[, shift]]] ) -> img
```

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

Draws a circle.

The function `cv::circle` draws a simple or filled circle with a given center and radius.

**Parameters**

- img** Image where the circle is drawn.
- center** Center of the circle.
- radius** Radius of the circle.
- color** Circle color.
- thickness** Thickness of the circle outline, if positive. Negative values, like `FILLED`, mean that a filled circle is to be drawn.
- lineType** Type of the circle boundary. See `LineTypes`
- shift** Number of fractional bits in the coordinates of the center and in the radius value.

**Examples:**

`samples/cpp/convexhull.cpp`, `samples/cpp/falsecolor.cpp`, `samples/cpp/kmeans.cpp`, `samples/cpp/lkdemo.cpp`,  
`samples/cpp/minarea.cpp`, `samples/cpp/tutorial_code/imgProc/basic_drawing/Drawing_1.cpp`,  
`samples/cpp/tutorial_code/imgProc/basic_drawing/Drawing_2.cpp`, `samples/cpp/tutorial_code/imgTrans/houghcircles.cpp`, and  
`samples/dnn/openpose.cpp`.

# 예제 6 – 마우스를 통한 상호작용

- Callback 함수를 활용

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 클릭
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
```



# 예제 7 – 마우스 드래그로 도형 크기 조절

- Callback 함수를 활용

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
```



# 예제 8 – 페인팅

- 페인팅 기능

프로그램 2-9

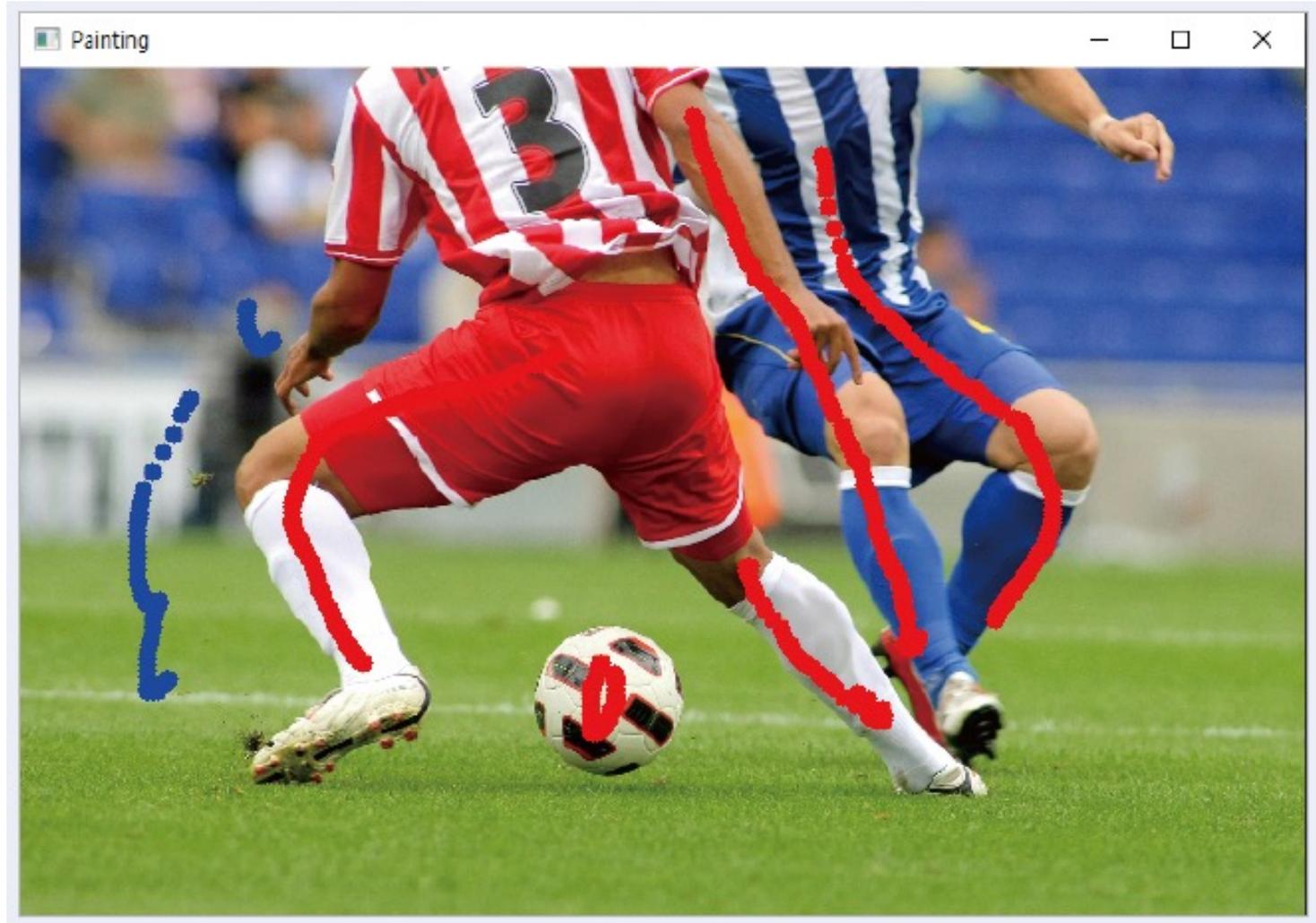
빨간색 붓과 파란색 붓으로 페인팅하기

```
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
```

```
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
```

## 예제 8 – 페인팅

- 페인팅 기능



**End of slide**

---