

파이썬으로 배우는 딥러닝

이미지 전처리 (OpenCV)

Ch.02 OpenCV기본기능 연습하기

[OpenCV 예제1] 이미지 파일 읽고, 표시하고, 저장하기

◆ Ch.02\W2-1.py에 아래 코드 작성해 넣기

```
1 import cv2 as cv          # OpenCV 라이브러리를 cv라는 이름으로 가져옴
2 import sys                # 시스템 종료 등의 기능을 위한 sys 모듈 가져옴
3
4 # 이미지 파일 읽기 (상대 경로 기준)
5 img = cv.imread('Ch.02/soccer.jpg')
6
7 # 이미지가 없는 경우 오류 메시지 출력 후 종료
8 if img is None:
9     sys.exit('File Not Found') # 이미지 파일을 찾을 수 없으면 프로그램 종료
10
11 # 이미지 화면에 출력
12 cv.imshow('Image Display', img) # 'Image Display'라는 이름의 창에 이미지 보여주기
13
14 # 이미지를 새로운 파일로 저장
15 cv.imwrite('Ch.02/soccer2.jpg', img) # soccer2.jpg 이름으로 이미지 저장
16
17 # 키 입력 대기
18 cv.waitKey() # 키보드 입력이 있을 때까지 대기
19
20 # 열려 있는 모든 OpenCV 창 닫기
21 cv.destroyAllWindows() # 창 닫기
```



[OpenCV 예제1] 이미지 파일 읽고, 표시하고, 저장하기

◆ **cv.imread(filename, flags)**

- 이미지 파일을 읽어와서 NumPy 배열 형식으로 반환하는 함수
- OpenCV에서 이미지 작업을 시작할때 가장 먼저 사용하는 필수 함수
- flags의 종류 : 생략시 기본값은 IMREAD_COLOR

◆ **cv.imshow(window_name, image)**

- OpenCV에서 이미지를 화면에 표시할 때 사용하는 함수

◆ **cv.imwrite(filename, image)**

- OpenCV에서 이미지를 파일로 저장할 때 사용하는 함수

[OpenCV 예제1] 이미지 파일을 읽고, 표시하고, 저장하기

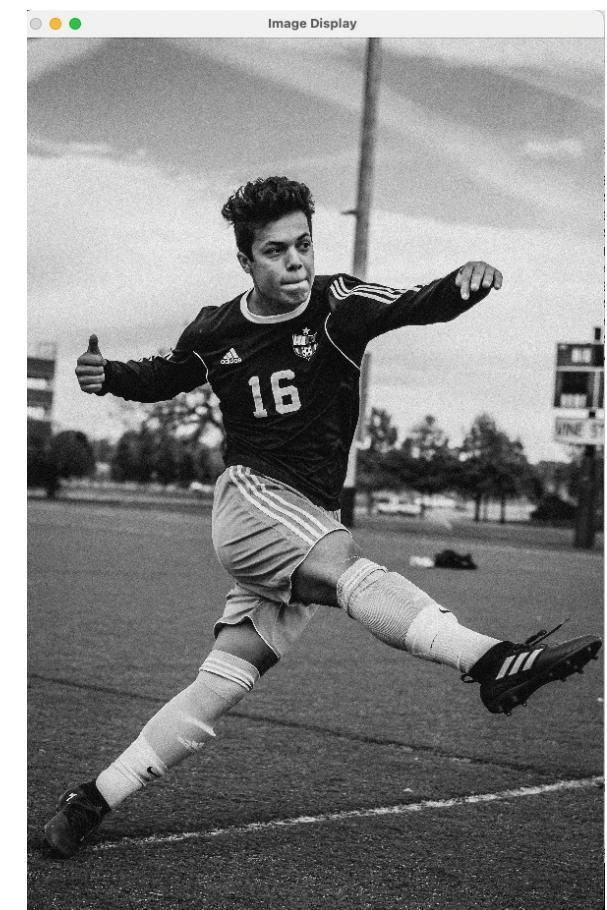
◆ cv.imread의 flags 종류

Flag 이름	정수값	설명
cv2.IMREAD_COLOR	1	컬러 이미지로 읽기 (기본값), 알파 채널 무시
cv2.IMREAD_GRAYSCALE	0	그레이스케일(흑백) 이미지로 읽기
cv2.IMREAD_UNCHANGED	-1	원본 그대로 읽기 (알파 채널 포함 등)
cv2.IMREAD_ANYDEPTH	2	16비트/32비트 이미지도 비트 깊이 유지하며 읽기
cv2.IMREAD_ANYCOLOR	4	가능한 경우 컬러로 읽기
cv2.IMREAD_REDUCED_COLOR_2	17	이미지 크기를 1/2로 줄여서 컬러로 읽기
cv2.IMREAD_REDUCED_COLOR_4	33	이미지 크기를 1/4로 줄여서 컬러로 읽기
cv2.IMREAD_REDUCED_COLOR_8	65	이미지 크기를 1/8로 줄여서 컬러로 읽기
cv2.IMREAD_REDUCED_GRAYSCALE_2	16	이미지 크기를 1/2로 줄여서 흑백으로 읽기
cv2.IMREAD_REDUCED_GRAYSCALE_4	32	이미지 크기를 1/4로 줄여서 흑백으로 읽기
cv2.IMREAD_REDUCED_GRAYSCALE_8	64	이미지 크기를 1/8로 줄여서 흑백으로 읽기

[OpenCV 예제1] 이미지 파일을 읽고, 표시하고, 저장하기

◆ API 응용 : cv.imread() 에 cv.IMREAD_GRAYSCALE 옵션 추가해보기

```
1 import cv2 as cv          # OpenCV 라이브러리를 cv라는 이름으로 가져옴
2 import sys                # 시스템 종료 등의 기능을 위한 sys 모듈 가져옴
3
4 # 이미지 파일 읽기 (상대 경로 기준)
5 img = cv.imread('Ch.02/soccer.jpg', cv.IMREAD_GRAYSCALE)
6
7 # 이미지가 없는 경우 오류 메시지 출력 후 종료
8 if img is None:
9     sys.exit('File Not Found') # 이미지 파일을 찾을 수 없으면 프로그램 종료
10
11 # 이미지 화면에 출력
12 cv.imshow('Image Display', img) # 'Image Display'라는 이름의 창에 이미지 보여주기
13
14 # 이미지를 새로운 파일로 저장
15 cv.imwrite('Ch.02/soccer2.jpg', img) # soccer2.jpg 이름으로 이미지 저장
16
17 # 키 입력 대기
18 cv.waitKey() # 키보드 입력이 있을 때까지 대기
19
20 # 열려 있는 모든 OpenCV 창 닫기
21 cv.destroyAllWindows() # 창 닫기
```



[OpenCV 예제2] 이미지 파일을 변환하기-cvtColor

◆ Ch.02\2-2.py에 아래 코드 작성해 넣기

```
1 import cv2 as cv          # OpenCV 라이브러리를 cv라는 이름으로 가져옴
2 import sys                # 시스템 종료 등의 기능을 위한 sys 모듈 가져옴
3
4 # 이미지 파일 읽기 (상대 경로 기준)
5 img = cv.imread('Ch.02/soccer.jpg')
6
7 # 이미지가 없는 경우 오류 메시지 출력 후 종료
8 if img is None:
9     sys.exit('File Not Found') # 이미지 파일을 찾을 수 없으면 프로그램 종료
10
11 # 기존 이미지를 Gray 이미지로 변환
12 Gray_image = cv.cvtColor(img, cv.COLOR_BGR2GRAY)
13
14 # 이미지 화면에 출력
15 cv.imshow('Image Display', Gray_image) # 'Image Display'라는 이름의 창에 이미지 보여주기
16
17 # 키 입력 대기
18 cv.waitKey() # 키보드 입력이 있을 때까지 대기
19
20 # 열려 있는 모든 OpenCV 창 닫기
21 cv.destroyAllWindows() # 창 닫기
```



[OpenCV 예제2] 이미지 파일을 변환하기-cvtColor

◆ cv.cvtColor(img, flags) API의 flag 종류

변환 목적	Flag 이름	설명
BGR → GRAY	<code>cv2.COLOR_BGR2GRAY</code>	컬러 이미지를 흑백으로 변환
BGR → RGB	<code>cv2.COLOR_BGR2RGB</code>	OpenCV는 기본이 BGR이므로 RGB로 바꾸는데 사용
BGR → HSV	<code>cv2.COLOR_BGR2HSV</code>	색상(Hue), 채도(Saturation), 명도(Value)로 분리
BGR → LAB	<code>cv2.COLOR_BGR2LAB</code>	밝기 기반 LAB 색 공간
BGR → YUV	<code>cv2.COLOR_BGR2YUV</code>	방송 등에서 쓰이는 색 공간
BGR → HLS	<code>cv2.COLOR_BGR2HLS</code>	Hue, Lightness, Saturation
GRAY → BGR	<code>cv2.COLOR_GRAY2BGR</code>	흑백 이미지를 컬러로 변환 (채널 늘리기)
RGB → HSV	<code>cv2.COLOR_RGB2HSV</code>	RGB 기반 HSV 변환
HSV → BGR	<code>cv2.COLOR_HSV2BGR</code>	HSV 이미지 → BGR 이미지로 복원

[OpenCV 예제3] 이미지 파일을 변환하기-resize

◆ Ch.02W2-3.py에 아래 코드 작성해 넣기

```
1 import cv2 as cv          # OpenCV 라이브러리를 cv라는 이름으로 가져옴
2 import sys                # 시스템 종료 등의 기능을 위한 sys 모듈 가져옴
3
4 # 이미지 파일 읽기 (상대 경로 기준)
5 img = cv.imread('Ch.02/soccer.jpg')
6
7 # 이미지가 없는 경우 오류 메시지 출력 후 종료
8 if img is None:
9     sys.exit('File Not Found') # 이미지 파일을 찾을 수 없으면 프로그램 종료
10
11 # 기존 이미지를 축소
12 Small_image = cv.resize(img, dsize=(0,0), fx=0.3, fy=0.3) # 줄여쓰기
13
14 # 이미지 화면에 출력
15 cv.imshow('Image Display', Small_image) # 'Image Display'라는 이름의 창에 이미지 보여주기
16
17 # 키 입력 대기
18 cv.waitKey() # 키보드 입력이 있을 때까지 대기
19
20 # 열려 있는 모든 OpenCV 창 닫기
21 cv.destroyAllWindows() # 창 닫기
```



[OpenCV 예제3] 이미지 파일을 변환하기-cvtColor

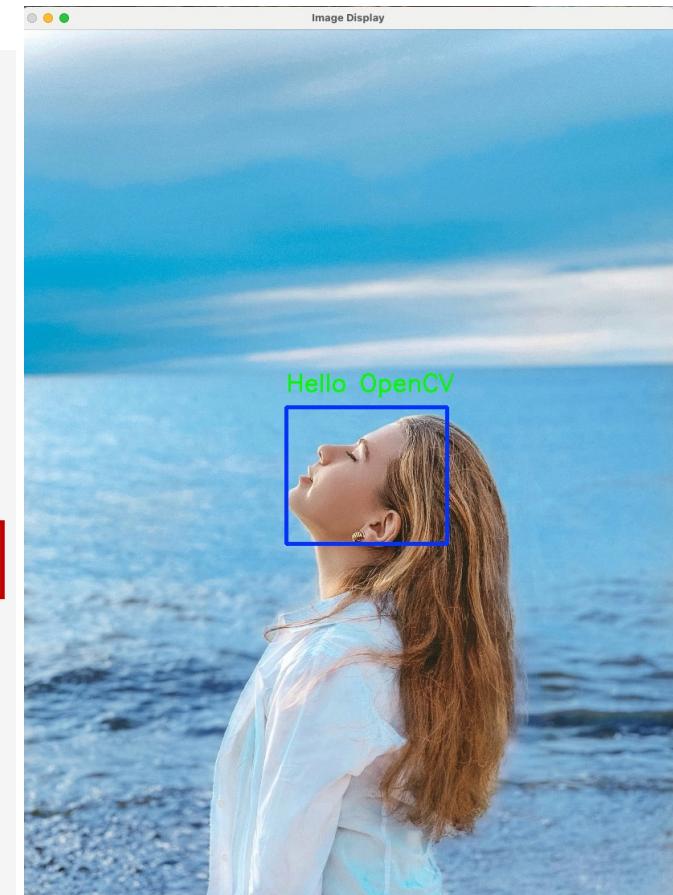
◆ cv.resize(img, dsize=(0,0), fx=0.3, fy=0.3)

- OpenCV를 사용하여 이미지를 가로와 세로 방향으로 각각 30%로 축소하는 코드
- img: 원본 이미지
- dsize=(0,0): 출력 이미지의 크기를 직접 지정하지 않고, 대신 비율로 크기를 조정하기 위해 (0,0)으로 설정
- fx=0.3, fy=0.3: 가로(fx)와 세로(fy) 방향으로 각각 0.3배로 크기를 축소.
- resized = cv.resize(img, (400, 300)) # 너비 400, 높이 300

[OpenCV 예제4] 이미지에 도형과 글씨 추가하기

◆ Ch.02W2-4.py에 아래 코드 작성해 넣기

```
1 import cv2 as cv          # OpenCV 라이브러리를 cv라는 이름으로 가져옴
2 import sys                # 시스템 종료 등의 기능을 위한 sys 모듈 가져옴
3
4 # 이미지 파일 읽기 (상대 경로 기준)
5 img = cv.imread('Ch.02/girl.jpg') // Line 5
6
7 # 이미지가 없는 경우 오류 메시지 출력 후 종료
8 if img is None:
9     sys.exit('File Not Found') # 이미지 파일을 찾을 수 없으면 프로그램 종료
10
11 # 이미지에 사각형 그리기 (좌상단 (330,470), 우하단 (530, 640), 파란색, 두께 3)
12 cv.rectangle(img, (330, 470), (530, 640), color=(255, 0, 0), thickness=3) // Line 12
13
14 # 이미지에 텍스트 쓰기 ("Hello OpenCV", 위치 (330, 450), 폰트, 크기 1, 초록색, 두께 2)
15 cv.putText(img, 'Hello OpenCV', org=(330, 450), fontFace=cv.FONT_HERSHEY_SIMPLEX,
16             fontScale=1, color=(0, 255, 0), thickness=2) // Line 15
17
18 # 이미지 화면에 출력
19 cv.imshow('Image Display', img) # 'Image Display'라는 이름의 창에 이미지 보여주기
20
21 # 키 입력 대기
22 cv.waitKey() # 키보드 입력이 있을 때까지 대기
23
24 # 열려 있는 모든 OpenCV 창 닫기
25 cv.destroyAllWindows() # 창 닫기
```



[OpenCV 예제4] 이미지에 도형과 글씨 추가하기

◆ cv.rectangle(img, (330, 470), (530, 640), (255, 0, 0), thickness=3)

- OpenCV를 사용하여 이미지에 직사각형을 그리는 코드
- img: 직사각형을 그릴 이미지
- (330, 470): 직사각형의 왼쪽 위 모서리 좌표
- (530, 640): 직사각형의 오른쪽 아래 모서리 좌표
- (255, 0, 0): 직사각형의 색상으로, BGR 형식으로 파란색을 지정.
- 3: 직사각형 테두리의 두께를 나타냅니다. 여기서는 3픽셀 두께로 직사각형이 그려짐

[OpenCV 예제4] 이미지에 도형과 글씨 추가하기

◆ `cv.putText(img, 'Hello OpenCV', org=(330, 450), fontFace=cv.FONT_HERSHEY_SIMPLEX, fontScale=1, color=(0, 255, 0), thickness=2)`

- OpenCV를 사용하여 이미지에 텍스트를 추가하는 코드
- img: 텍스트를 추가할 이미지
- 'Hello OpenCV': 이미지에 표시할 텍스트 내용.
- (330, 450): 텍스트를 그릴 시작 좌표. 이 좌표는 텍스트의 왼쪽 아래 모서리 위치.
- cv.FONT_HERSHEY_SIMPLEX: OpenCV에서 제공하는 글꼴 중 하나인 Hershey Simplex 글꼴을 사용
- 1: 텍스트 크기를 조정하는 스케일 팩터. 여기서는 글꼴 크기가 1로 설정
- (0, 255, 0): 텍스트 색상. BGR 형식으로 지정하며, (0, 255, 0)은 초록색을 나타냄
- 2: 텍스트 테두리의 두께. 여기서는 2픽셀 두께로 텍스트가 그려짐.

[OpenCV 예제5] 마우스를 이용하여 이미지에 사각형 그리기

◆ Ch.02W2-5.py에 아래 코드 작성해 넣기

```
1 import cv2 as cv
2
3 # 전역 변수 선언
4 drawing = False          # 마우스 클릭 상태 저장 (드래그 중인지 여부)
5 start_point = (-1, -1)    # 시작점
6 end_point = (-1, -1)      # 끝점
7
8 # 마우스 콜백 함수 정의
9 def draw_rectangle(event, x, y, flags, param):
10     global drawing, start_point, end_point, img, temp_img
11
12     if event == cv.EVENT_LBUTTONDOWN:
13         drawing = True
14         start_point = (x, y)
15         end_point = (x, y)
16         temp_img = img.copy() # 드래그 시작 시 이미지 복사
17
18     elif event == cv.EVENT_MOUSEMOVE:
19         if drawing:
20             end_point = (x, y)
21             temp_img = img.copy()
22             cv.rectangle(temp_img, start_point, end_point, (255, 0, 0), 2)
23
24     elif event == cv.EVENT_LBUTTONUP:
25         drawing = False
26         end_point = (x, y)
27         cv.rectangle(img, start_point, end_point, (255, 0, 0), 2)
28         temp_img = img.copy()
```



[OpenCV 예제5] 마우스를 이용하여 이미지에 사각형 그리기

◆ Ch.02\W2-5.py에 아래 코드 작성해 넣기

```
29  # 이미지 로드
30  img = cv.imread('Ch.02/girl.jpg')
31  if img is None:
32      print('이미지 파일을 찾을 수 없습니다.')
33      exit()
34
35
36  temp_img = img.copy()
37
38  # 윈도우 생성 및 마우스 콜백 등록
39  cv.namedWindow('Draw Rectangle')
40  cv.setMouseCallback('Draw Rectangle', draw_rectangle)
41
42  # 루프를 돌면서 이미지 표시
43  while True:
44      cv.imshow('Draw Rectangle', temp_img)
45      key = cv.waitKey(1) & 0xFF
46      if key == 27: # ESC 키를 누르면 종료
47          break
48
49  cv.destroyAllWindows()
50 
```



[OpenCV 예제5] 마우스를 이용하여 이미지에 사각형 그리기

◆ `cv.setMouseCallback('Draw Rectangle', draw_rectangle)`

- OpenCV에서 특정 창(윈도우)에 마우스 이벤트 콜백 함수를 설정하는 코드
- 'Draw Rectangle': 마우스 이벤트를 설정할 창(윈도우)의 이름. OpenCV 창을 생성할 때 지정한 이름이 들어감.
- `draw_rectangle`: 마우스 이벤트가 발생할 때 호출될 콜백 함수. 여기서 `draw_rectangle`는 사용자가 정의한 함수로, 마우스 클릭, 드래그 등의 이벤트가 발생할 때 해당 함수가 실행됨.

[OpenCV 예제5] 마우스를 이용하여 이미지에 사각형 그리기

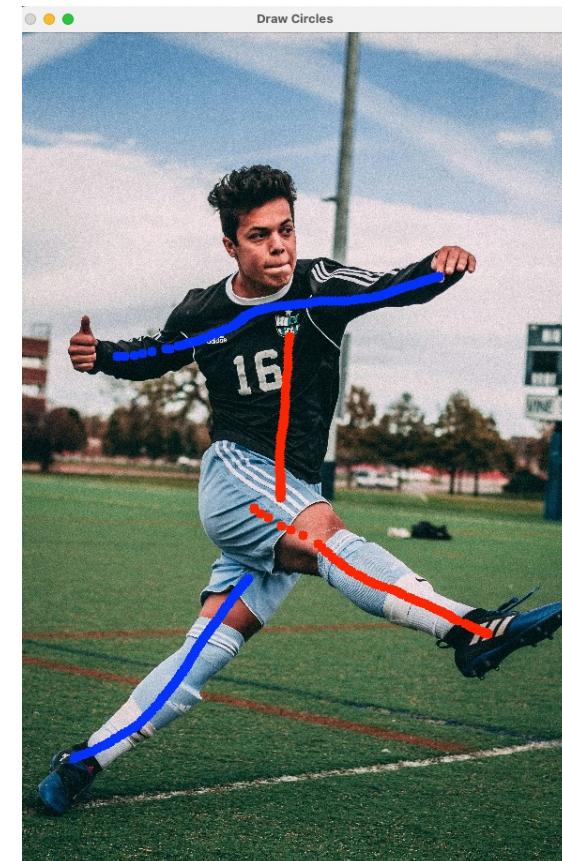
◆ Mouse event 종류

이벤트 상수	설명
<code>cv.EVENT_MOUSEMOVE</code>	마우스가 이동할 때
<code>cv.EVENT_LBUTTONDOWN</code>	왼쪽 버튼을 누를 때
<code>cv.EVENT_RBUTTONDOWN</code>	오른쪽 버튼을 누를 때
<code>cv.EVENT_MBUTTONDOWN</code>	가운데 버튼 (휠)을 누를 때
<code>cv.EVENT_LBUTTONUP</code>	왼쪽 버튼을 뗄 때
<code>cv.EVENT_RBUTTONUP</code>	오른쪽 버튼을 뗄 때
<code>cv.EVENT_MBUTTONUP</code>	가운데 버튼을 뗄 때
<code>cv.EVENT_LBUTTONDOWNDBLCLK</code>	왼쪽 버튼 더블 클릭
<code>cv.EVENT_RBUTTONDOWNDBLCLK</code>	오른쪽 버튼 더블 클릭
<code>cv.EVENT_MBUTTONDOWNDBLCLK</code>	가운데 버튼 더블 클릭

[OpenCV 예제6] 마우스를 이용하여 이미지에 페인팅 해보기

◆ Ch.02W2-6.py에 아래 코드 작성해 넣기

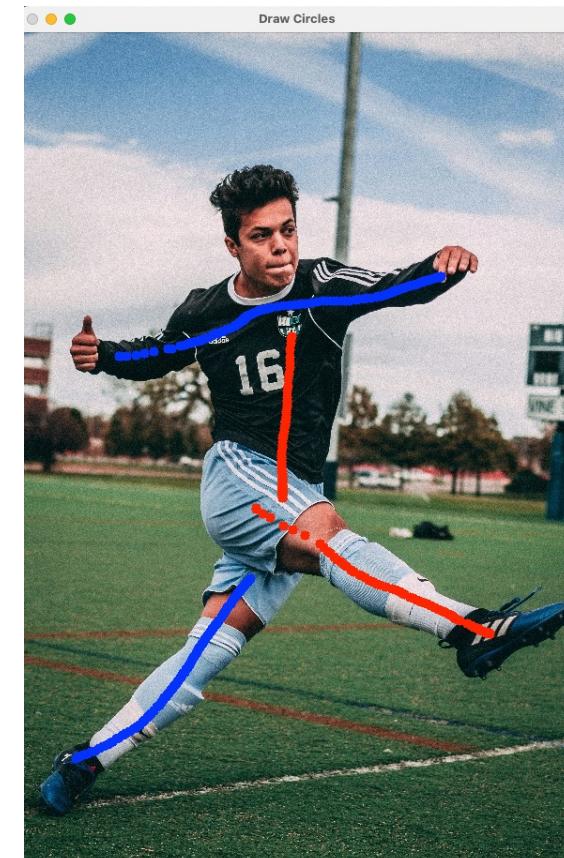
```
1 import cv2 as cv
2
3 # 마우스 이벤트 콜백 함수
4 def draw(event, x, y, flags, param):
5     global drawing_left, drawing_right, img
6
7     # 왼쪽 버튼 클릭 → 드래그 시작
8     if event == cv.EVENT_LBUTTONDOWN:
9         drawing_left = True
10
11     # 왼쪽 버튼 드래그 중
12     elif event == cv.EVENT_MOUSEMOVE and drawing_left:
13         cv.circle(img, (x, y), 5, (255, 0, 0), -1) # 파란색 원
14
15     # 왼쪽 버튼 드래그 종료
16     elif event == cv.EVENT_LBUTTONUP:
17         drawing_left = False
18
19     # 오른쪽 버튼 클릭 → 드래그 시작
20     elif event == cv.EVENT_RBUTTONDOWN:
21         drawing_right = True
22
23     # 오른쪽 버튼 드래그 중
24     elif event == cv.EVENT_MOUSEMOVE and drawing_right:
25         cv.circle(img, (x, y), 5, (0, 0, 255), -1) # 빨간색 원
```



[OpenCV 예제6] 마우스를 이용하여 이미지에 페인팅 해보기

◆ Ch.02W2-6.py에 아래 코드 작성해 넣기

```
27      # 오른쪽 버튼 드래그 종료
28      elif event == cv.EVENT_RBUTTONDOWN:
29          drawing_right = False
30
31  # 이미지 준비
32  img = cv.imread('Ch.02/soccer.jpg')
33  if img is None:
34      print('이미지를 불러올 수 없습니다.')
35      exit()
36
37  # 드래그 상태 플래그
38  drawing_left = False
39  drawing_right = False
40
41  # 창 생성 및 마우스 콜백 연결
42  cv.namedWindow('Draw Circles')
43  cv.setMouseCallback('Draw Circles', draw)
44
45  # 루프를 돌며 이미지 출력
46  while True:
47      cv.imshow('Draw Circles', img)
48      key = cv.waitKey(1) & 0xFF
49      if key == 27: # ESC 키
50          break
51
52  cv.destroyAllWindows()
```



[OpenCV 예제6] 마우스를 이용하여 이미지에 페인팅 해보기

◆ **cv.circle(img, (x, y), BrushSiz, LColor, -1)**

- OpenCV를 사용하여 원을 그리는 코드
- img: 원을 그릴 이미지
- (x, y): 원의 중심 좌표. 여기서 (x, y)는 마우스 이벤트로 전달된 좌표로, 마우스 위치에 원이 그려짐.
- BrushSiz: 원의 반지름을 나타내는 변수로, 여기서는 브러시 크기처럼 사용됨.
- LColor: 원의 색상을 나타내는 변수로, BGR 형식으로 지정. 예를 들어 (255, 0, 0)은 파란색
- -1: 이 값은 원을 채우는 방식. -1은 원을 채운 상태로 그린다는 의미. 양의 값(예: 1, 2 등)은 테두리만 그릴 때 사용하는 두께를 지정하는 값

[OpenCV 예제7] 웹캠 촬영 영상을 모니터에 표시하기

◆ Ch.02W2-7.py에 아래 코드 작성해 넣기

```
1 import cv2 as cv
2
3 # 웹캠 열기 (기본 장치는 0번)
4 cap = cv.VideoCapture(0)
5
6 # 웹캠 열기에 실패했는지 확인
7 if not cap.isOpened():
8     print("웹캠을 열 수 없습니다.")
9     exit()
10
11 while True:
12     ret, frame = cap.read() # 한 프레임 읽기
13     if not ret:
14         print("프레임을 가져올 수 없습니다. 종료합니다.")
15         break
16
17     # 프레임을 창에 출력
18     cv.imshow('Live Video', frame)
19
20     # 키보드에서 'q'를 누르면 루프 종료
21     if cv.waitKey(1) & 0xFF == ord('q'):
22         break
23
24     # 사용한 자원 해제
25     cap.release()
26     cv.destroyAllWindows()
```



[OpenCV 예제7] 웹캠 촬영 영상을 모니터에 표시하기

◆ **cap = VideoCapture(0)**

- OpenCV에서 카메라(or 동영상 파일)을 불러오기 위한 함수
- 첫번째 옵션 '0' → 장치 인덱스
- (생략된) 두번째 옵션 → cv.CAP_DSHOW는 DirectShow 사용명시(Windows용)
- 두번째 옵션을 명시적으로 설정하면 안정성을 높임

[OpenCV 예제7] 웹캠 촬영 영상을 모니터에 표시하기

◆ cap = VideoCapture(0, cv.CAP_DSHOW) 외 다양한 옵션들

Flag	설명	운영체제
cv2.CAP_ANY	자동으로 적절한 백엔드 선택	공통
cv2.CAP_V4L2	Video4Linux2 (Linux)	Linux
cv2.CAP_V4L	Video4Linux (구버전)	Linux
cv2.CAP_DSHOW	DirectShow	Windows
cv2.CAP_MSMF	Microsoft Media Foundation	Windows
cv2.CAP_AVFOUNDATION	AVFoundation	macOS
cv2.CAP_QT	QtKit (구버전 macOS)	macOS
cv2.CAP_FFMPEG	FFmpeg (동영상 파일용)	공통
cv2.CAP_GSTREAMER	GStreamer 사용	Linux
cv2.CAP_INTEL_MFX	Intel Media SDK	Windows
cv2.CAP_GPHOTO2	DSLR 카메라 등	Linux

[OpenCV 예제8] 웹캠에서 캡처한 영상을 메모리에 저장하기

◆ Ch.02W2-8.py에 아래 코드 작성해 넣기

```
1  import cv2 as cv
2
3  cap = cv.VideoCapture(0) # 웹캠 열기
4  if not cap.isOpened():
5      print("웹캠을 열 수 없습니다.")
6      exit()
7
8  captured_images = [] # 캡처된 이미지 저장용 리스트
9
10 print("▶ 'a' 키: 한 장 캡처, 'q' 키: 캡처 이미지 보기 및 종료")
11
12 while True:
13     ret, frame = cap.read()
14     if not ret:
15         print("프레임을 가져올 수 없습니다. 종료합니다.")
16         break
17
18     cv.imshow('Live Webcam', frame)
19     key = cv.waitKey(1) & 0xFF
20
21     if key == ord('a'):
22         # 현재 프레임을 복사해서 저장
23         captured_images.append(frame.copy())
24         print(f"📸 {len(captured_images)}번째 이미지 캡처됨")
25
26     elif key == ord('q'):
27         print(f"🖼️ 총 {len(captured_images)}장의 이미지가 캡처되었습니다.")
28         break
29
30     cap.release()
31     cv.destroyAllWindows()
32
33     # 캡처된 이미지들을 하나씩 보여주기
34     for idx, img in enumerate(captured_images):
35         cv.imshow(f'Captured Image {idx+1}', img)
36         cv.waitKey(0) # 아무 키 누르면 다음 이미지로
37
38     cv.destroyAllWindows()
```

[OpenCV 예제8] 웹캠에서 캡쳐한 영상을 메모리에 저장하기

◆ Ch.02W2-8.py에 아래 코드 작성해 넣기

