Computer Vision

Lecture 03 OpenCV 기초

황선희

동양미래대학교
DONGYANG MIRAE UNIVERSITY

프로그래밍 실습 문제 1 - 답안

```python
import torch

# 크기 3x4의 무작위 텐서 생성
tensor = torch.rand(3, 4)

# 두 번째 열을 0으로 설정
tensor[:, 1] = 0

# 모든 원소의 합 계산
sum_all_elements = torch.sum(tensor)

print("Tensor:\n", tensor)
print("Sum of all elements:", sum_all_elements.item())
```

프로그래밍 실습 문제 2 - 답안

```python
import torch

# 크기 5x5의 무작위 텐서 생성
tensor = torch.rand(5, 5)

# 조건부 필터링 (0.5보다 큰 값들만 추출)
filtered_tensor = tensor[tensor > 0.5]

print("Original Tensor:\n", tensor)
print("Filtered Tensor (values > 0.5):\n", filtered_tensor)
```

프로그래밍 실습 문제 3 - 답안

```
import numpy as np
import torch

# 1. 크기 4x4의 NumPy 배열 생성
np_array = np.random.rand(4, 4)
print("Original NumPy Array:\n", np_array)

# 2. NumPy 배열을 PyTorch 텐서로 변환
torch_tensor = torch.from_numpy(np_array)
print("Converted PyTorch Tensor:\n", torch_tensor)

# 3. PyTorch 텐서를 'torch_tensor.pt' 파일로 저장
torch.save(torch_tensor, 'torch_tensor.pt’)

# 4. 저장된 파일을 불러와 텐서로 변환
loaded_tensor = torch.load('torch_tensor.pt')

# 원본 텐서와 불러온 텐서가 동일한지 확인
print("Loaded PyTorch Tensor:\n", loaded_tensor)
print("Are they equal?", torch.equal(torch_tensor, loaded_tensor))
```

목차

1. OpenCV 소개
2. 프로그래밍 환경 설정
3. 이미지 읽고 화면에 표시하기
4. 이미지 형태와 크기 변환하기
5. 비디오 읽기
6. 그래픽 기능과 사용자 인터페이스 만들기

1. OpenCV 소개

- OpenCV (Open Source Computer Vision)
  - 원 저자는 인텔로, 실시간 컴퓨터 비전을 목적으로 개발된 라이브러리
  - 클래스와 함수는 C와 C++로 개발되었으며, 전체 코드는 180만 라인 이상으로 구성
  - 인터페이스 언어는 C, C++, 자바, 자바스크립트, 파이썬
  - 지원하는 OS 플랫폼은 윈도우, 리눅스, macOS, 안드로이드, iOS로 다양함
  - 교육과 상업 목적 모두 무료 (Apache 2 License)

https://opencv.org/

1. OpenCV 소개

- Free Tutorial
  - https://opencv.org/university/free-opencv-course/?utm_source=opcv&utm_medium=menu&utm_campaign=obc

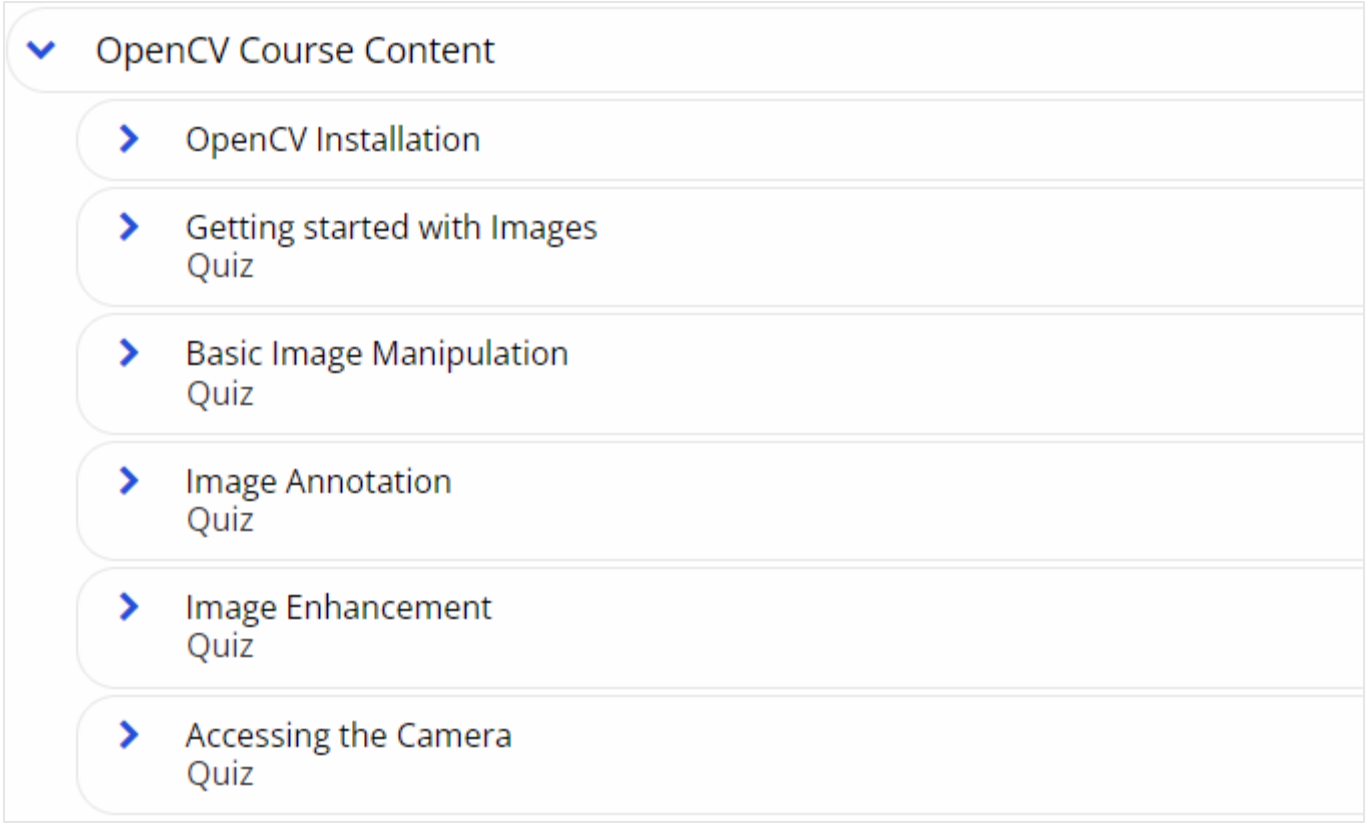
What's covered in this course?

Module 1 : Getting Started With Images
Module 2 : Basic Image Manipulation
Module 3 : Histograms and Color Segmentation
Module 4 : Video Processing and Analysis
Module 5 : Contour and Shape Analysis
Module 6 : Playing Games Using CV (HCI)
Module 7 : Building and Deploying Web Apps with Streamlit

Module 08: Image Registration Techniques
Module 09: ArUco Markers for Augmented Reality
Module 10: Deep Learning with OpenCV
Module 11: Face and Landmark Detection
Module 12: Object Detection
Module 13: Object Tracking
Module 14: Human Pose Estimation

1. OpenCV 소개

- Free Tutorial
  - https://opencv.org/university/free-opencv-course/?utm_source=opcv&utm_medium=menu&utm_campaign=obc



2. 프로그래밍 환경 설정

- Python과 함께 pip 설치
  - 설치 옵션에서 주의 깊게 확인할 것

- (Optional) Conda환경 설정
  - Anaconda 설치 (Window 외 기타 OS는 관련 페이지를 확인해야 함)
    - https://docs.anaconda.com/anaconda/install/windows/
  - Anaconda Prompt에 다음 명령어를 입력하여 가상환경 생성
    - conda create -n vision
    - conda activate vision
    - (가상환경 해제) conda deactivate

- OpenCV 설치
  - CMD창에 다음 pip 명령어로 설치 (Linux 계열 PC 사용 시, Terminal에서 명령어로 설치)
    - pip install opencv-python

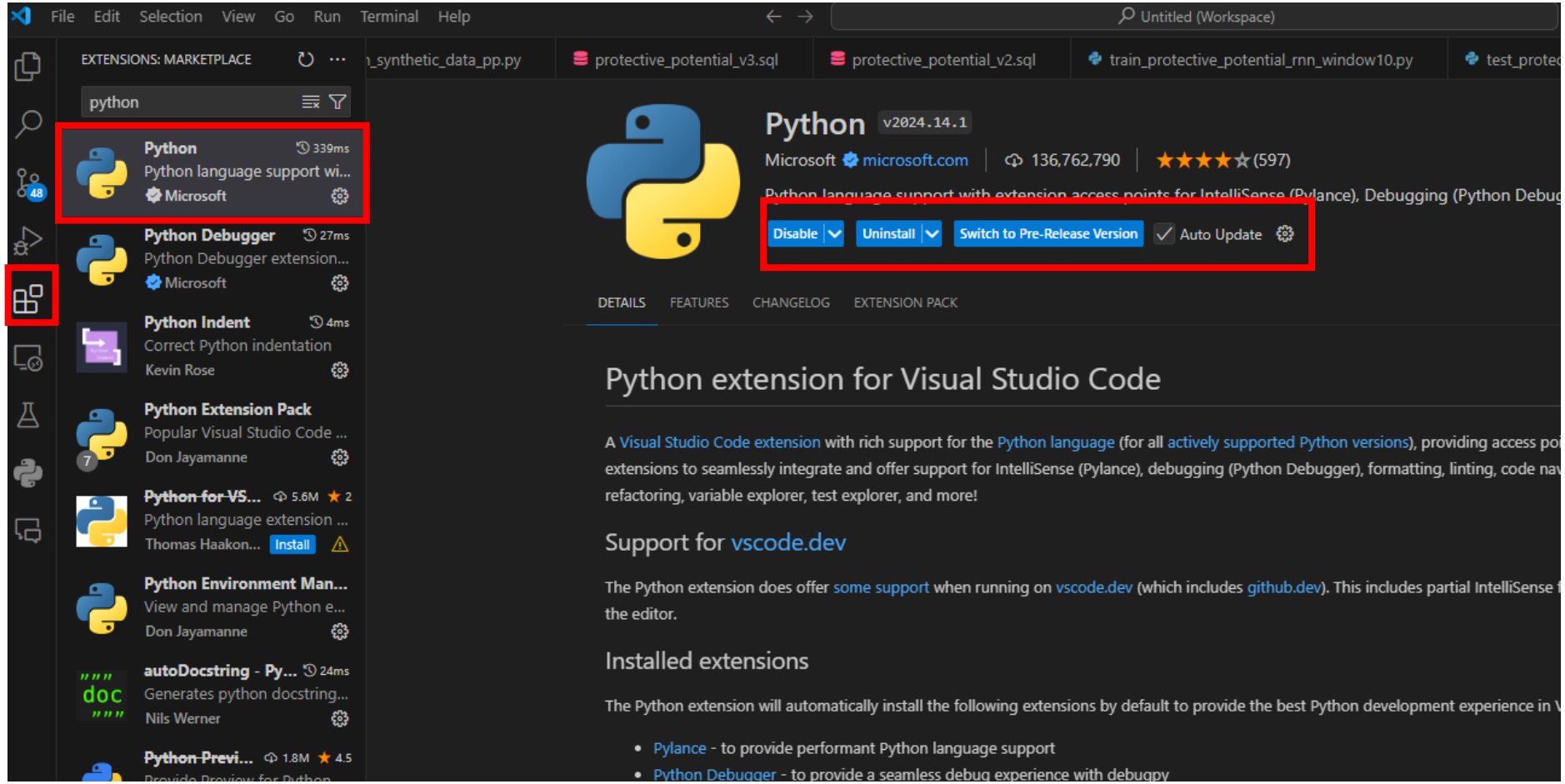
2. 프로그래밍 환경 설정

- Anaconda Prompt 또는 CMD 창에서 python 실행하여 다음 명령어가 작동되는지 확인

```
import cv2
print(cv2.__version__)
```

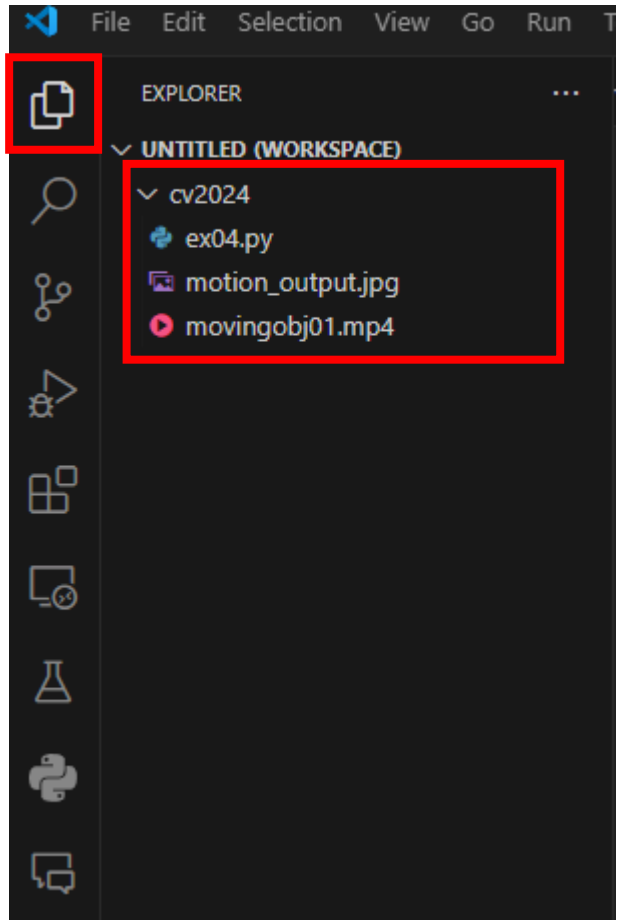
2. 프로그래밍 환경 설정

- VS Code 환경설정
  - Python extension 설치 (Install 버튼 클릭)



2. 프로그래밍 환경 설정

- VS Code 환경설정
  - 소스코드와 데이터가 저장된 폴더를 드래그하여, 연결
    → Ctrl + Shift + p 버튼을 입력 → Python : Select Interpreter 클릭
    → (여러 폴더가 Open상태인 경우) 데이터 및 소스코드를 저장할 폴더 선택
    → Python (또는 Conda) 선택
    → 소스코드 작성 및 Ctrl + F5 키로 실행



3. 이미지 읽고 화면에 표시하기

- 이미지 파일을 읽고 화면에 띄우기
  - 데이터 불러오기: cv2.imread('파일명')
  - 데이터 화면에 띄우기: cv2.imshow('창이름', 데이터명)
  - 화면 멈추기: cv2.waitKey(0) - 키 입력을 기다리는 대기함수 (0; 무한대기)
  - 모든 창 닫기: cv2.destroyAllWindows() - 키보드 입력 값 발생 시 창 종료됨

cv2024 > lec03.py > ...

```python
import cv2
import sys

img = cv2.imread('soccer.jpg')

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

cv2.imshow('Image', img)

cv2.waitKey(0)
cv2.destroyAllWindows()
```

3. 이미지 읽고 화면에 표시하기

- 이미지 데이터의 타입과 형태
  - 이미지 타입: Numpy Array
  - 이미지 형태: 3차원 데이터, (height = 행의 수, width = 열의 수, channels = 컬러정보)
    - 채널이란? 이미지의 색상 정보를 표현하는 차원 (그레이스케일=1, 컬러=3)
    - 컬러 채널은 blue, green, red 순으로 구성 (bgr)

```python
import cv2
import sys

img = cv2.imread('soccer.jpg')

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

print(type(img))
print(img.shape)
```

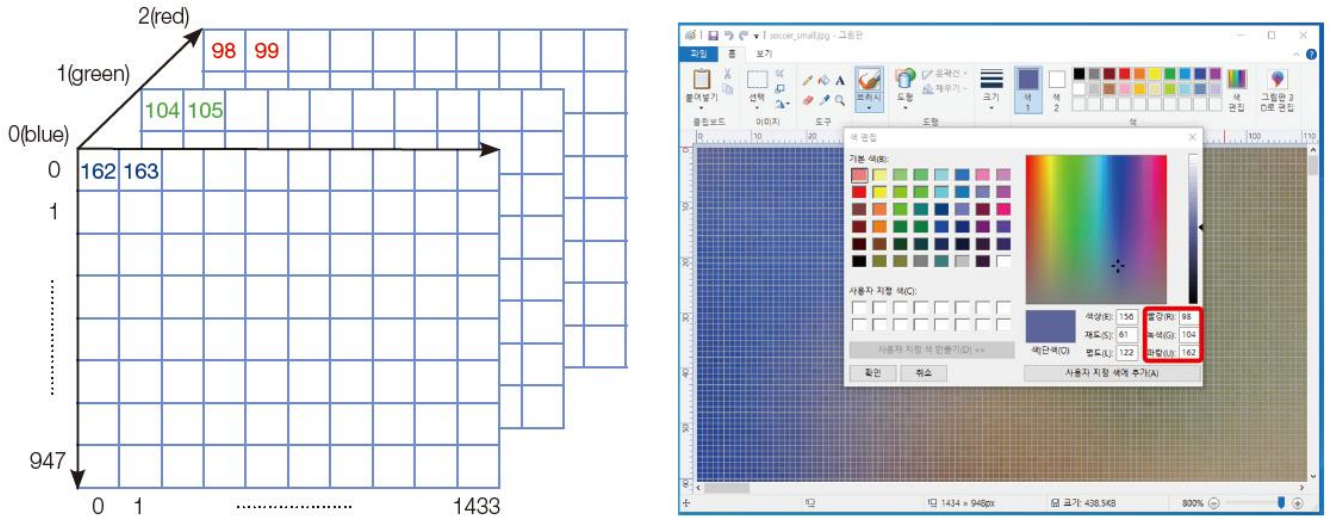
```
<class 'numpy.ndarray'>
(396, 600, 3)
```

3. 이미지 읽고 화면에 표시하기

- 영상 화소의 표현
  - 모든 픽셀의 값은 0~255 (2^8, 8비트로 표현) 내 하나의 값으로 표현됨
  - 인덱싱과 슬라이싱 모두 가능 (Ndarray)

```
print(img[0,0,0], img[0,0,1], img[0,0,2])
print(img[0,1,0], img[0,1,1], img[0,1,2])
```

```
169 104 82
169 104 82
```



(a) 프로그램으로 조사 (b) 그림판으로 조사

그림 2-9 img 객체가 표현하는 영상의 구조와 내용

3. 이미지 읽고 화면에 표시하기

- 이미지 데이터를 복사하기
  - 데이터.copy() 함수로 복사

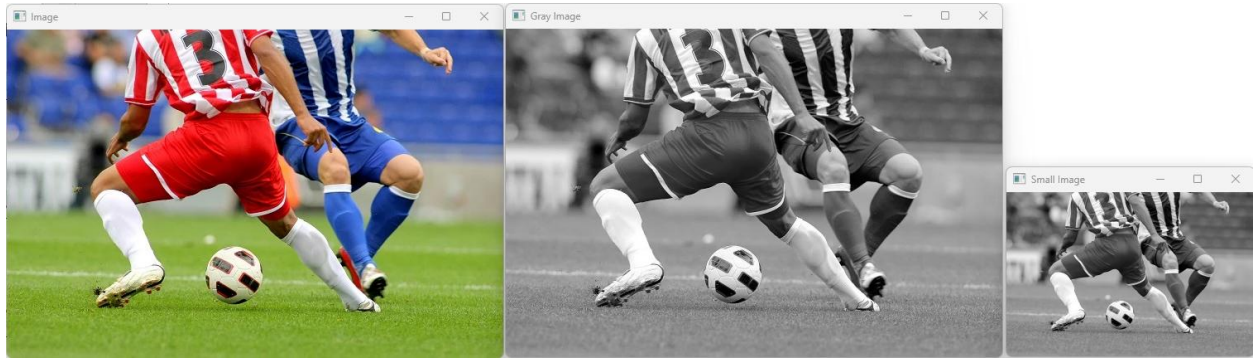
```
import cv2
img = cv2.imread('soccer.jpg')
img2 = img.copy()

# 동일한지 비교
print(img.all() == img2.all())
```

```
True
```

4. 이미지 형태와 크기 변환하기

- 이미지를 그레이스케일로 변환, 축소(0.5배), 저장하기

```python
import cv2
import sys

img = cv2.imread('soccer.jpg')

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

# 이미지 그레이 스케일로 변환
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

# 이미지 크기 절반으로 축소 0.5배
img_small = cv2.resize(gray, (0, 0), fx=0.5, fy=0.5)

cv2.imshow('Image', img)
cv2.imshow('Gray Image', gray)
cv2.imshow('Small Image', img_small)

cv2.imwrite('gray.jpg', gray)
cv2.imwrite('gray_small.jpg', img_small)

cv2.waitKey(0)
cv2.destroyAllWindows()
```



4. 이미지 형태와 크기 변환하기

- cvtColor 함수가 컬러 영상을 명암 영상으로 바꾸는 방법
  - 다음 계산식을 활용하여, 데이터를 변환 (RGB 컬러 도메인 별 비율이 상이함, round는 반올림)

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

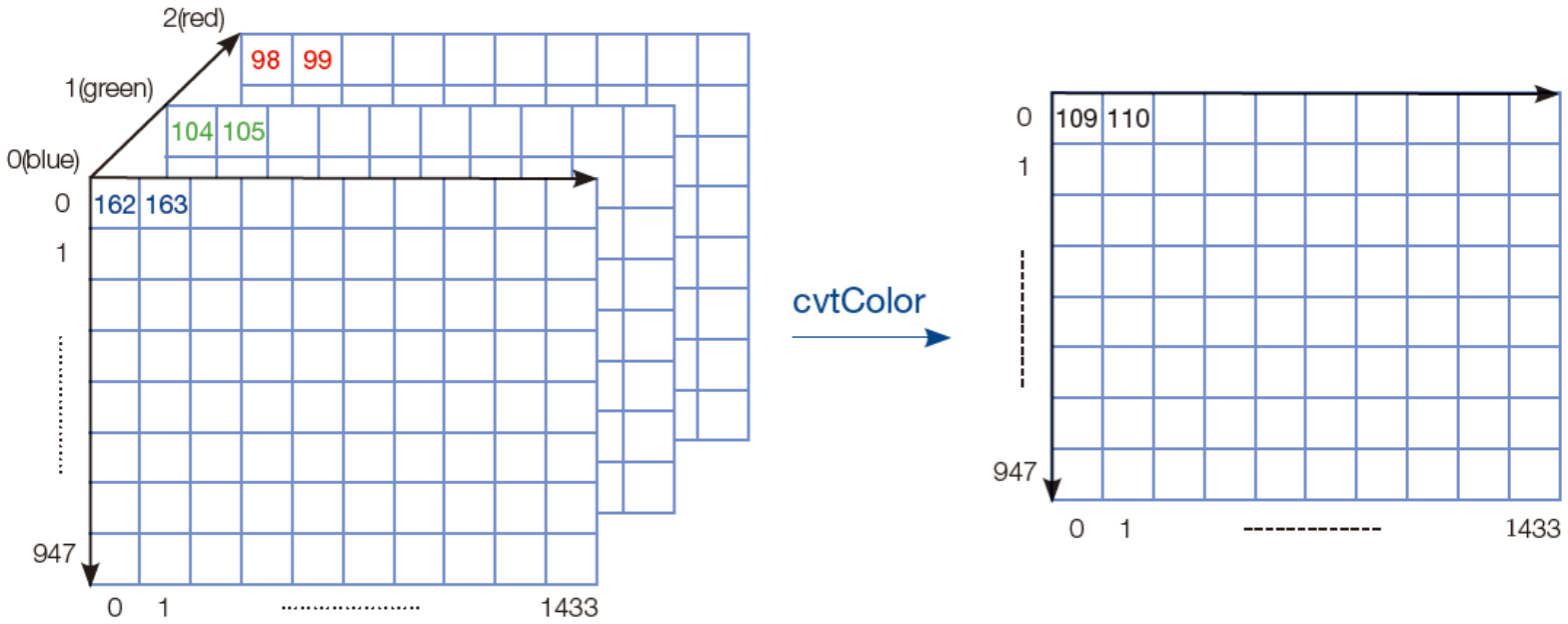
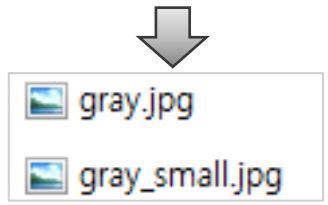


그림 2-10 BGR 컬러 영상을 명암 영상으로 변환

4. 이미지 형태와 크기 변환하기

- 이미지를 그레이스케일로 변환, 축소(0.5배), 저장하기
  - 컬러변환: cv2.cvtColor(데이터, 변환컬러)
  - 크기변환: cv2.resize(데이터, (w, h)) 또는
    cv2.resize(데이터, (0,0), fx=비율, fy=비율)
  - 이미지 저장: cv2.imwrite('파일명(타입포함)', 데이터)

```python
import cv2
import sys

img = cv2.imread('soccer.jpg')

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

# 이미지 그레이 스케일로 변환
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

# 이미지 크기 절반으로 축소 0.5배
img_small = cv2.resize(gray, (0, 0), fx=0.5, fy=0.5)

cv2.imshow('Image', img)
cv2.imshow('Gray Image', gray)
cv2.imshow('Small Image', img_small)

cv2.imwrite('gray.jpg', gray)
cv2.imwrite('gray_small.jpg', img_small)

cv2.waitKey(0)
cv2.destroyAllWindows()
```



4. 이미지 형태와 크기 변환하기

- 이미지 덮어쓰기, 원본과의 차이 확인, 마스크 이미지 생성하기
  - 데이터[index] = 값 형태로 덮어쓰기 가능
  - 이미지 차이 계산: cv2.absdiff(데이터1, 데이터2)
  - 이미지 차이에 따른 이진화: cv2.threshod(차이, 기준값, 변환(흰색 255), 유형(이진화))

```
import cv2

img = cv2.imread('soccer.jpg')
img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
img2 = img.copy()
img2[50:100, 50:100] = 255

# 이미지 차이 계산
diff = cv2.absdiff(img, img2)
# 차이가 있는 영역을 찾아서 이진화
_, mask = cv2.threshold(diff, 30, 255, cv2.THRESH_BINARY)

cv2.imshow('img', img)
cv2.imshow('img2', img2)
cv2.imshow('diff', diff)
cv2.imshow('mask', mask)

cv2.waitKey(0)
cv2.destroyAllWindows()
```

4. 이미지 형태와 크기 변환하기

- 마스크(임의생성)를 통해 배경(soccer) 이미지와, 전경(soccer_ball) 이미지를 합성하기
  - 마스크 데이터 생성: np.zeros(크기, 데이터타입) → 임의영역에 255 할당
  - 이진이미지 반전: cv2.bitwise_not(이진이미지) → 255, 0 영역이 서로 반전됨
  - 마스크 영역에 이미지 붙여넣기: cv2.bitwise_and(데이터, 붙여넣을 데이터, mask=영역)
  - 이미지 더하기: cv2.add(데이터1, 데이터2)

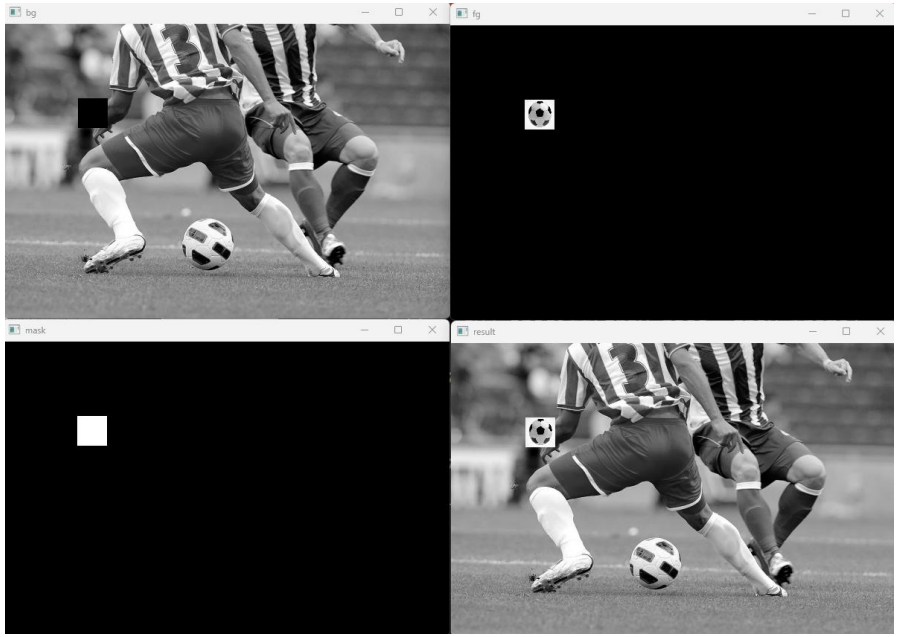
```python
import cv2
import numpy as np
bg = cv2.imread('soccer.jpg')
bg = cv2.cvtColor(bg, cv2.COLOR_BGR2GRAY)

ball = cv2.imread('soccer_ball.jpg')
ball = cv2.cvtColor(ball, cv2.COLOR_BGR2GRAY)
ball = cv2.resize(ball, (40,40))

fg = np.zeros(bg.shape, dtype=np.uint8)
fg[100:140, 100:140] = ball

# bg 크기의 binary image 생성하고 임의의 위치에 mask 생성
mask = np.zeros(bg.shape, dtype=np.uint8)
mask[100:140, 100:140] = 255
mask_inv = cv2.bitwise_not(mask)

bg = cv2.bitwise_and(bg, bg, mask=mask_inv)
fg = cv2.bitwise_and(fg, fg, mask=mask)

result = cv2.add(bg, fg)

cv2.imshow('mask', mask)
cv2.imshow('bg', bg)
cv2.imshow('fg', fg)
cv2.imshow('result', result)

cv2.waitKey(0)
cv2.destroyAllWindows()
```

soccer.jpg

soccer_ball.jpg

bg

fg



mask

result

5. 비디오 읽기

- PC에 연결된 웹 캠을 이용하여, 비디오 열기
  - 웹 캠 열기: cv2.VideoCapture(0, cv2.CAP_DSHOW)
  - 반복문을 통해, 연속 영상을 입력: ret, frame = cap.read()

```python
import cv2
import sys

cap = cv2.VideoCapture(0, cv2.CAP_DSHOW)

if not cap.isOpened():
    sys.exit('카메라를 열 수 없습니다.')

while True:
    ret, frame = cap.read()

    if not ret:
        print('비디오 읽기 오류')
        break

    cv2.imshow('video display', frame)

    # q가 입력되면 종료
    if cv2.waitKey(1) == ord('q'):
        break

cap.release()
cv2.destroyAllWindows()
```

5. 비디오 읽기

- 저장된 비디오 파일을 재생하기
  - 웹 캠 열기: cv2.VideoCapture('파일명')
  - 반복문을 통해, 연속 영상을 입력: ret, frame = cap.read()

```python
import cv2
import sys

# 비디오 파일 열기
cap = cv2.VideoCapture('movingobj01.mp4')

# 비디오 파일이 열리지 않은 경우
if not cap.isOpened():
    sys.exit('비디오 파일을 열 수 없습니다.')

# 비디오 파일 재생
while True:
    ret, frame = cap.read()
    if not ret:
        break

    cv2.imshow('video display', frame)

    # q를 입력하면 종료
    if cv2.waitKey(1) == ord('q'):
        break

cap.release()
cv2.destroyAllWindows()
```

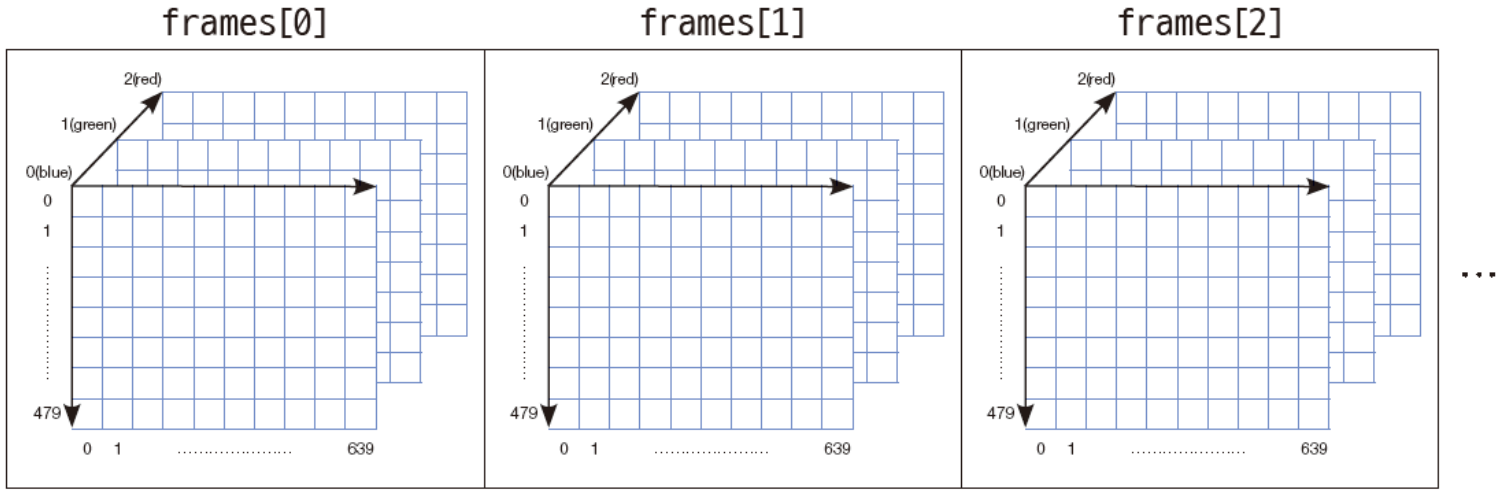
5. 비디오 읽기

- N초마다 한번씩 프레임 캡처하여, 가로로 이어붙이기
  - 초당 촬영된 이미지 수 불러오기(fps;frame per second): fps = cap.get(cv2.CAP_PROP_FPS)
  - 이미지 가로로 이어붙이기: cv2.hconcat(연속데이터, h;horizontal, 세로 cv2.vconcat, v;vertical)

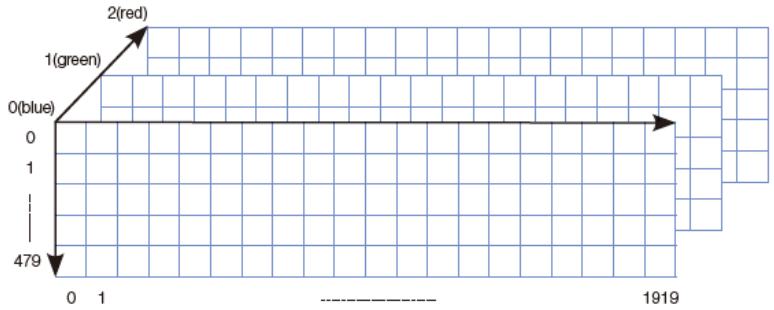
```python
import cv2
import sys
cap = cv2.VideoCapture('movingobj01.mp4')
if not cap.isOpened():
    sys.exit('비디오 파일을 열 수 없습니다.')

fps = cap.get(cv2.CAP_PROP_FPS)
frames = []
frame_count = 0
while True:
    ret, frame = cap.read()
    if not ret:
        break
    # 2초마다 한 번씩 프레임을 저장
    if frame_count % (fps*2) == 0:
        frames.append(frame)
    if len(frames) == 3:
        break
    frame_count += 1
cap.release()

# 세장의 이미지를 가로로 이어붙이기
result = cv2.hconcat(frames)
cv2.imshow('result', result)
cv2.waitKey(0)
cv2.destroyAllWindows()

```

5. 비디오 읽기

- 연속적으로 이어 붙인 데이터의 shape



(a) frames 리스트



(b) imgs 배열

그림 2-11 [프로그램 2-5]의 자료 구조

```
print(frames[0] .shape)
print(result.shape)
```

```
(720, 1280, 3)
(720, 3840, 3)
```

6. 그래픽 기능과 사용자 인터페이스 만들기

- 이미지에 도형을 그리고 글씨 쓰기
  - 사각형 그리기: cv2.rectangle(데이터, (시작좌표), (끝좌표), (컬러), 두께)
  - 글씨 쓰기: cv2.putText(데이터, '문구', (시작좌표), 글씨체, 크기, (컬러), 두께)
  - 컬러: (B, G, R) 0~255 사이의 값으로 할당

```python
import cv2

img = cv2.imread('soccer.jpg')
if img is None:
    print('이미지를 읽을 수 없습니다.')
    exit()

# 직사각형 그리기
cv2.rectangle(img, (50, 50), (150, 150), (0, 255, 0), 2)

# 글씨 쓰기
cv2.putText(img, 'Soccer', (50, 45), cv2.FONT_HERSHEY_SIMPLEX, 0.8, (0, 255, 0), 2)

cv2.imshow('Image', img)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

6. 그래픽 기능과 사용자 인터페이스 만들기

- 마우스를 이용하여, 선택된 좌표의 위치에 사각형을 그리고 글씨 작성하기
  - 왼쪽버튼 누를 때: cv2.EVENT_LBUTTONDOWN (뗄 때는 DOWN대신 UP으로 표현)
  - onMouse 함수 내 cv2.imshow의 역할: imshow로 나타난 화면에, 업데이트된 이미지를 덮는 역할
    - cv2.imshow가 두 번 등장한 이유

```python
import cv2

img = cv2.imread('soccer.jpg')

if img is None:
    print('이미지를 읽을 수 없습니다.')
    exit()

def onMouse(event, x, y, flags, param):
    global ix, iy
    if event == cv2.EVENT_LBUTTONDOWN:
        ix, iy = x, y
    elif event == cv2.EVENT_RBUTTONDOWN:
        cv2.rectangle(img, (ix, iy), (x, y), (0, 255, 0), 2)
        cv2.putText(img, 'soccer ball', (ix, iy), cv2.FONT_HERSHEY_SIMPLEX, 2, (0, 0, 255), 2)
    cv2.imshow('img', img)

cv2.imshow('img', img)
cv2.setMouseCallback('img', onMouse)

cv2.waitKey(0)
cv2.destroyAllWindows()
```

6. 그래픽 기능과 사용자 인터페이스 만들기

- 마우스를 이용하여, 선택된 좌표의 위치에 도형을 연속해서 그리기
  - 크기 1인 circle을 연속적으로 그리기 (cv2.circle)

```python
import cv2

img = cv2.imread('soccer.jpg')

brush_size = 1
def onMouse(event, x, y, flags, param):
    if event == cv2.EVENT_LBUTTONDOWN:
        cv2.circle(img, (x, y), brush_size, (255, 0, 0), 3)
        cv2.imshow('img', img)
    if event == cv2.EVENT_RBUTTONDOWN:
        cv2.circle(img, (x, y), brush_size, (0, 0, 255), 3)
        cv2.imshow('img', img)
    # 버튼을 누르고 이동하면
    if event == cv2.EVENT_MOUSEMOVE and flags == cv2.EVENT_FLAG_LBUTTON:
        cv2.circle(img, (x, y), brush_size, (255, 0, 0), 3)
        cv2.imshow('img', img)
    if event == cv2.EVENT_MOUSEMOVE and flags == cv2.EVENT_FLAG_RBUTTON:
        cv2.circle(img, (x, y), brush_size, (0, 0, 255), 3)
        cv2.imshow('img', img)
    cv2.imshow('img', img)

cv2.imshow('img', img)
cv2.setMouseCallback('img', onMouse)

cv2.waitKey(0)
cv2.destroyAllWindows()

```

6. 그래픽 기능과 사용자 인터페이스 만들기

- 이미지에 도형과 글씨 작성하기
  - 동그라미: cv2.circle(데이터, (좌표), 크기, (색), 두께)
  - 선: cv2.line(데이터, (시작좌표), (끝좌표), (색), 두께)
  - 타원: cv2.ellipse(데이터, (중심좌표), (축길이;가로반지름, 세로반지름), 시작각도, 끝각도, (색상), 두께)

```python
import cv2

img = cv2.imread('soccer.jpg')

# 동그라미 그리기
cv2.circle(img, (100, 100), 50, (0, 0, 255), 3)
# 선 그리기
cv2.line(img, (200, 200), (300, 300), (0, 255, 0), 3)
# 사각형 그리기
cv2.rectangle(img, (350, 350), (450, 450), (255, 0, 0), 3)
# 타원 그리기
cv2.ellipse(img, (150, 150), (100, 50), 0, 0, 360, (0, 255, 255), 3)

# 글씨 쓰기
cv2.putText(img, 'soccer ball', (100, 200), cv2.FONT_HERSHEY_SIMPLEX, 1, (255, 255, 255), 3)

cv2.imshow('img', img)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

프로그래밍 실습을 위한 동영상 촬영하기

- 2~3인이 한 팀으로 촬영할 것 (혼자 촬영하기 어려움)
- 카메라는 고정된 상태로 촬영
- 첫 프레임은 물체가 없는 상태를 촬영 (배경)
- 이후 프레임에는 물체(사람, 사물, 동물 등)가 움직이는 형태가 보이도록 촬영 (3초 이상)

프로그래밍 실습 문제 4

- 직접 촬영한 동영상 파일을 이용하여, 움직임이 포착된 모션샷 이미지 생성 및 저장하기
  - 제작에 필요한 opencv 함수: VideoCapture, copy, cvtColor, absdiff, threshold, bitwise_not, bitwise_and, add, imwrite
  - 제작과정: 배경이미지 설정 → 이미지 컬러변환 (이미지 차이 계산을 위해)
    → 이미지 차이 계산 (배경, 현재프레임) → 차이가 있는 영역을 합성 → 모션 이미지 업데이트

프로그래밍 실습 문제 4

```
import cv2

# 동영상 파일 또는 이미지 시퀀스에서 프레임 불러오기
cap = cv2.VideoCapture('movingobj01.mp4')

# 첫 번째 프레임을 읽고 배경으로 설정
ret, base_frame = cap.read()
if not ret:
    print("비디오를 읽을 수 없습니다.")
    cap.release()
    exit()

# 시퀀스 사진의 기본 베이스 이미지 생성 (초기 프레임)
sequence_image = base_frame.copy()
gray_base = cv2.cvtColor(base_frame, cv2.COLOR_BGR2GRAY)


...
```