

OpenCV-Python 시작하기

컴퓨터비전소개

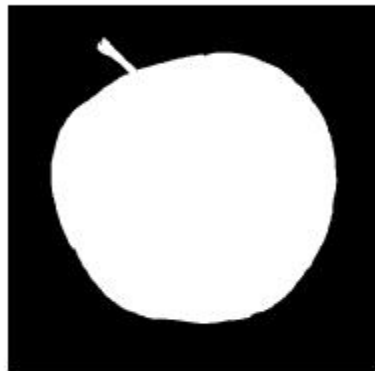
컴퓨터비전

✓ 컴퓨터비전(Computer vision)

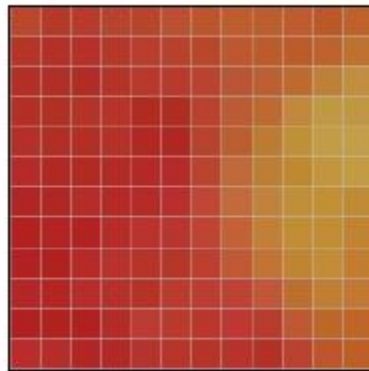
- 컴퓨터를 이용하여 정지 영상 또는 동영상으로부터 의미있는 정보를 추출하는 방법을 연구하는 학문
- 즉, 사람이 눈으로 사물을 보고 인지하는 작업을 컴퓨터가 수행하게끔 만드는 학문



사과?



둥글다?



빨간색이다?



꼭지 모양?



사과가 몇 개??

컴퓨터 비전

✓ 컴퓨터 비전과 영상처리(image processing)



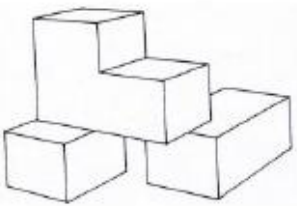
- 영상 처리는 영상을 입력으로 받아 화질을 개선하는 등의 처리를 하여 다시 영상을 출력으로 내보내는 작업
- 영상 처리는 컴퓨터 비전을 위한 전처리 작업

- 영상 처리는 영상을 다루는 모든 학문과 응용을 통틀어 지칭
- 컴퓨터 비전은 영상 인식과 같은 고수준의 영상 처리를 지칭

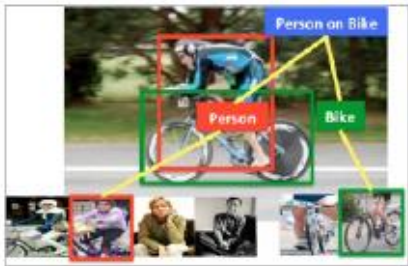
컴퓨터 비전 \approx 영상 처리

컴퓨터 비전의 역사

1966년 MIT “The Summer Vision Project” 연구가 컴퓨터 비전의 시초



IMAGENET



1960'

1970'

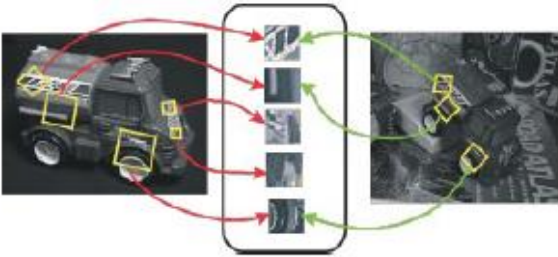
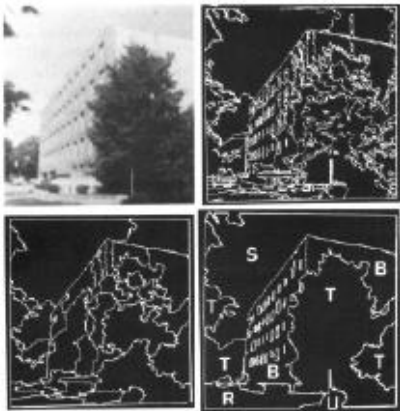
...

2000'

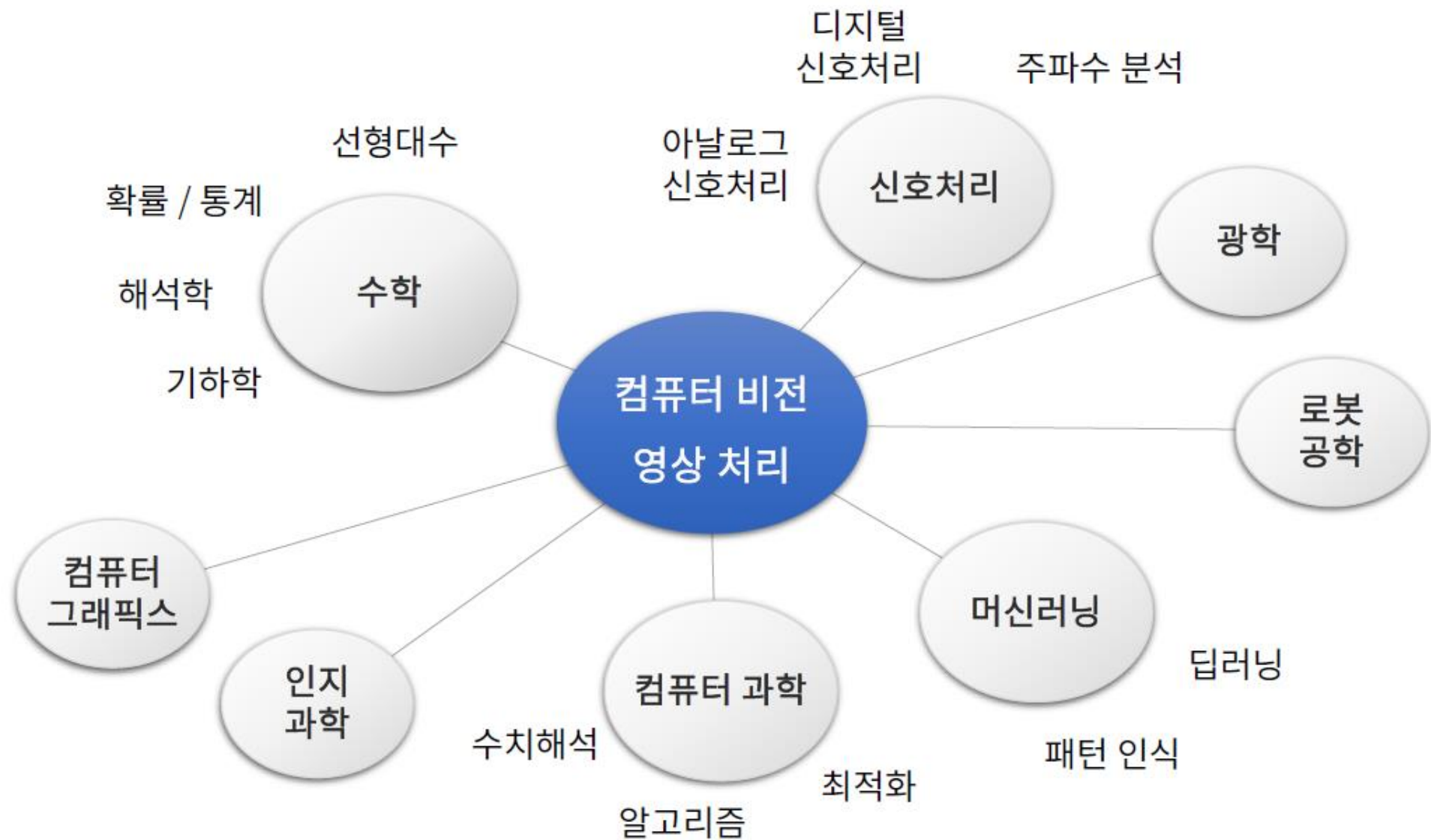
2010'

아날로그 방식의 사진 편집

위성으로부터 전송 받은 달 표면 사진의 화질 복원

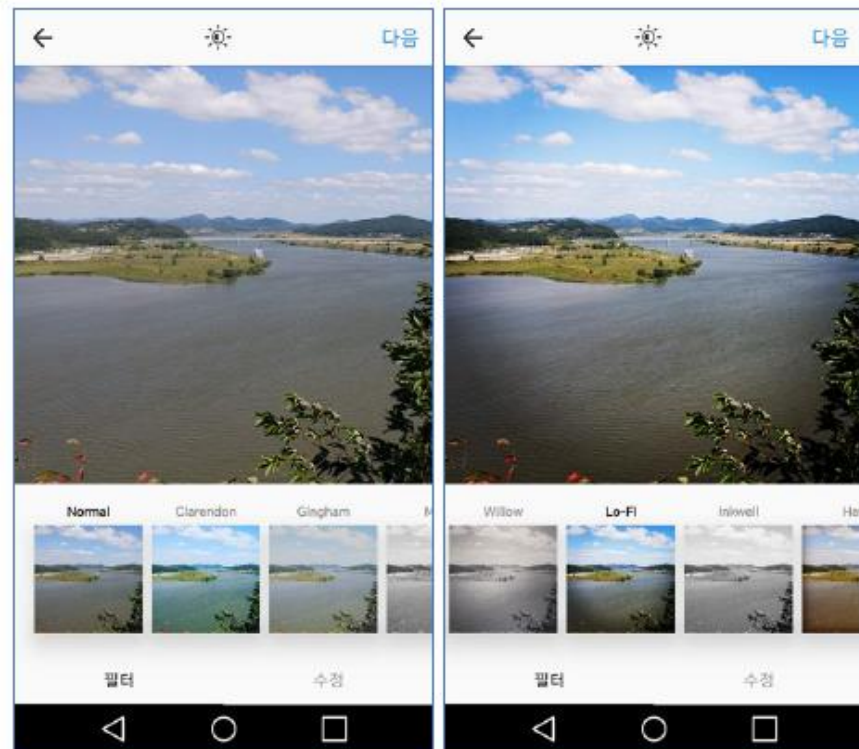


컴퓨터 비전 관련 분야

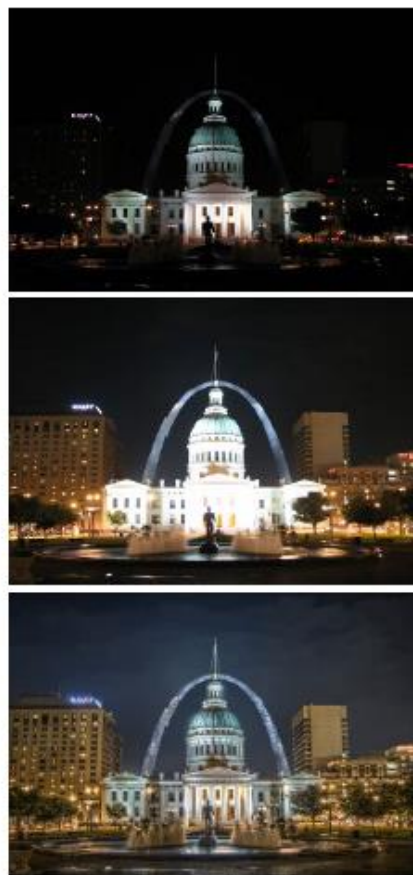


컴퓨터 비전 연구 분야

✓ 영상의 화질 개선



Filtering App



HDR



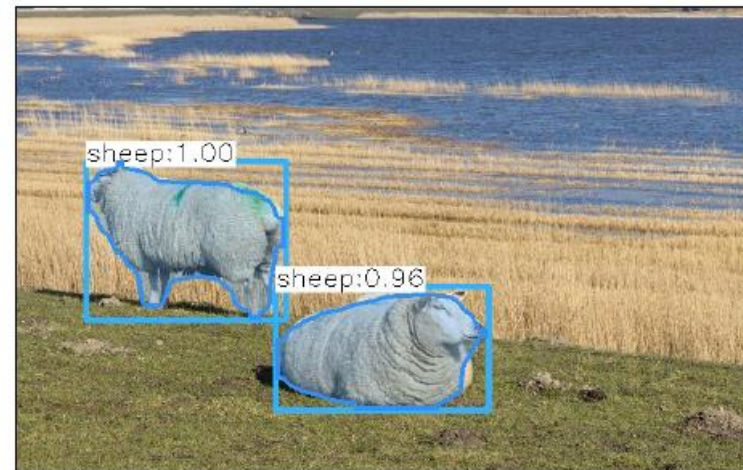
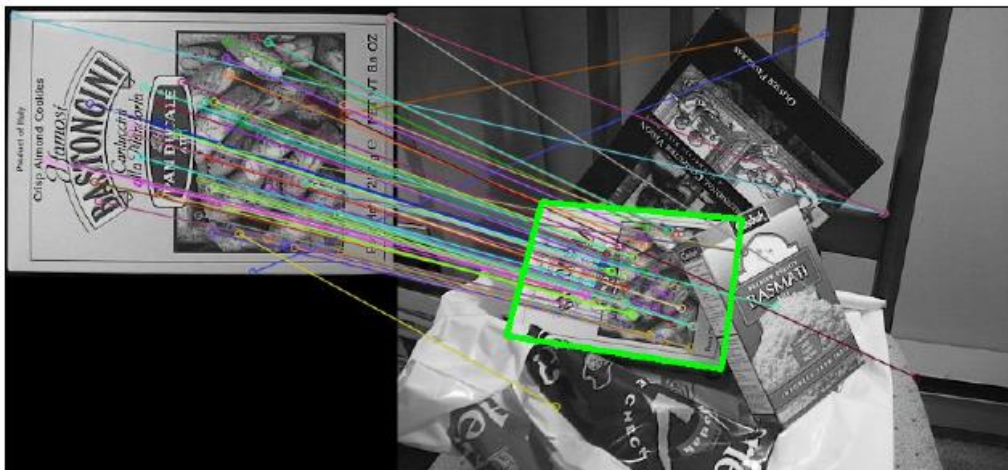
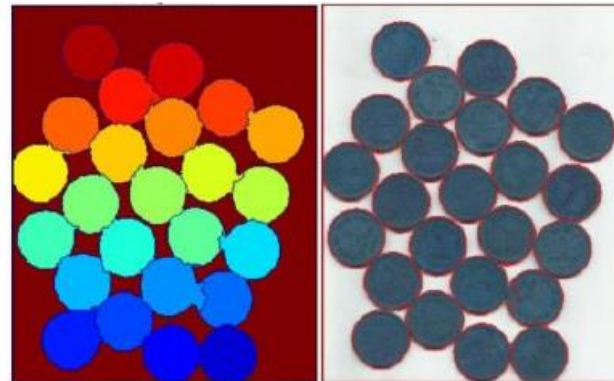
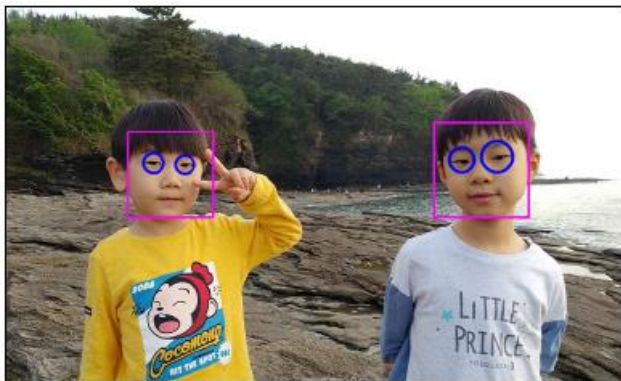
Image Noise Reduction



Super Resolution

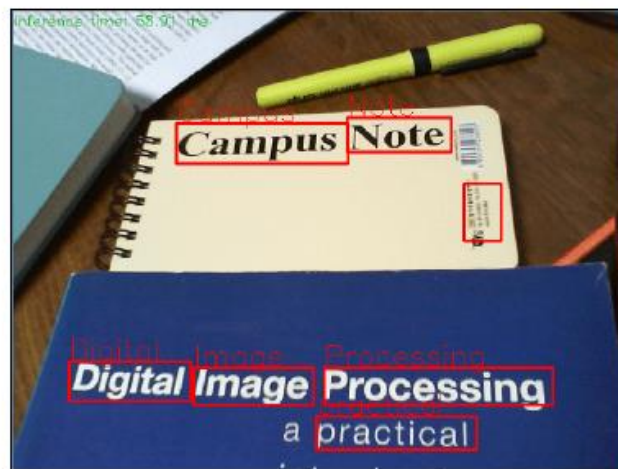
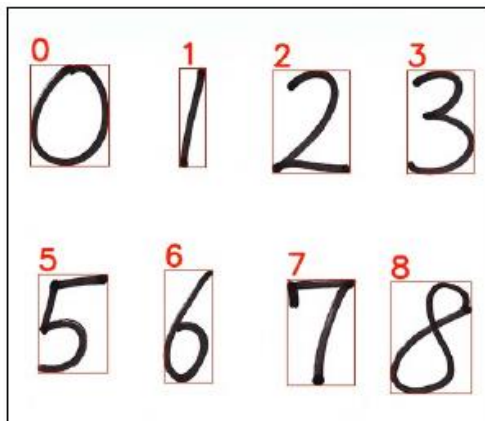
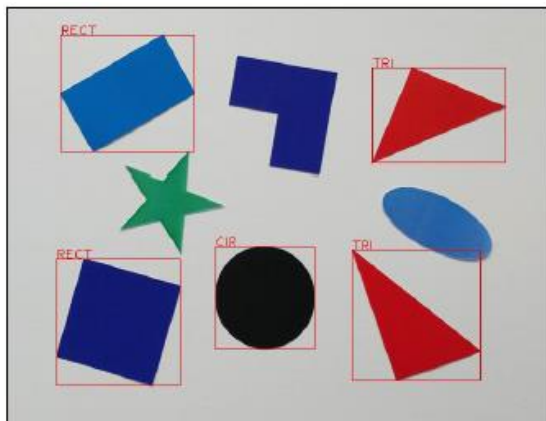
컴퓨터 비전 연구 분야

✓ 객체 검출(Object detection)과 영상 분할



컴퓨터 비전 연구 분야

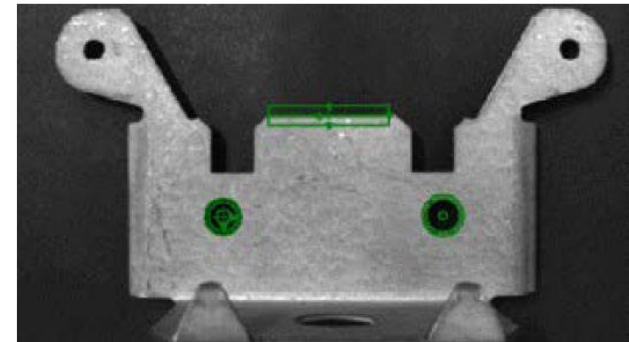
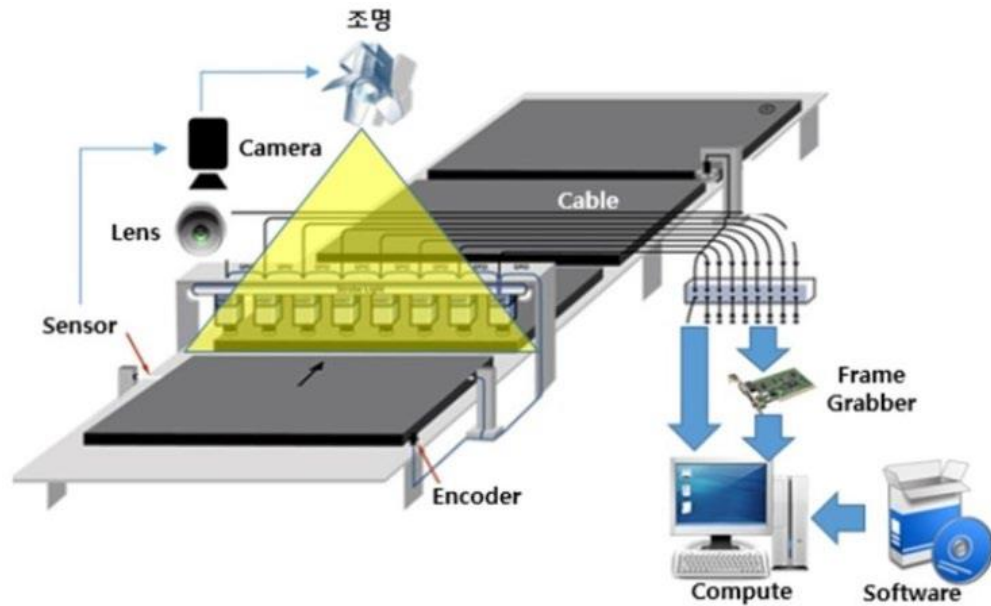
✓ 인식(Recognition)



컴퓨터 비전 응용 분야

✓ 머신 비전(machine vision)

- 공장 자동화: 제품의 불량 검사, 위치 확인, 측정 등
- 높은 정확도와 빠른 처리 시간 요구
- 조명, 렌즈, 필터, 실시간(Real-time) 처리



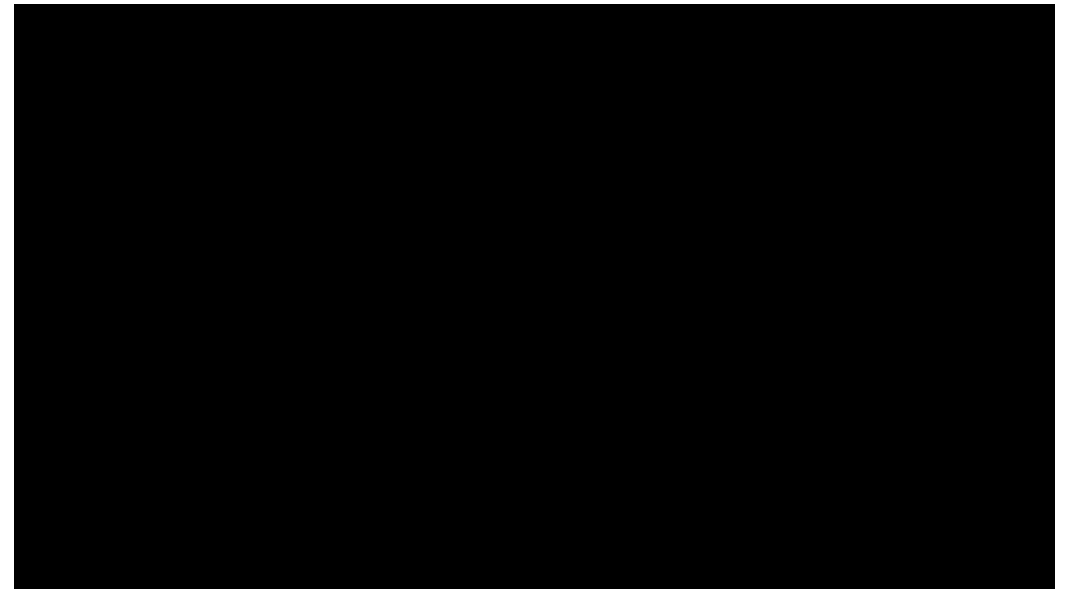
컴퓨터 비전 응용 분야

✓ 인공지능 서비스

- 입력 영상을 객체와 배경으로 분할 → 객체와 배경 인식 → 상황 인식 → 로봇과 자동차의 행동 지시
- Computer Vision + Sensor Fusion + Deep Learning
- 인공지능 로봇, Amazon Go, 구글/테슬라의 자율 주행 자동차



<https://youtu.be/NrmMk1Myrxc?t=26>

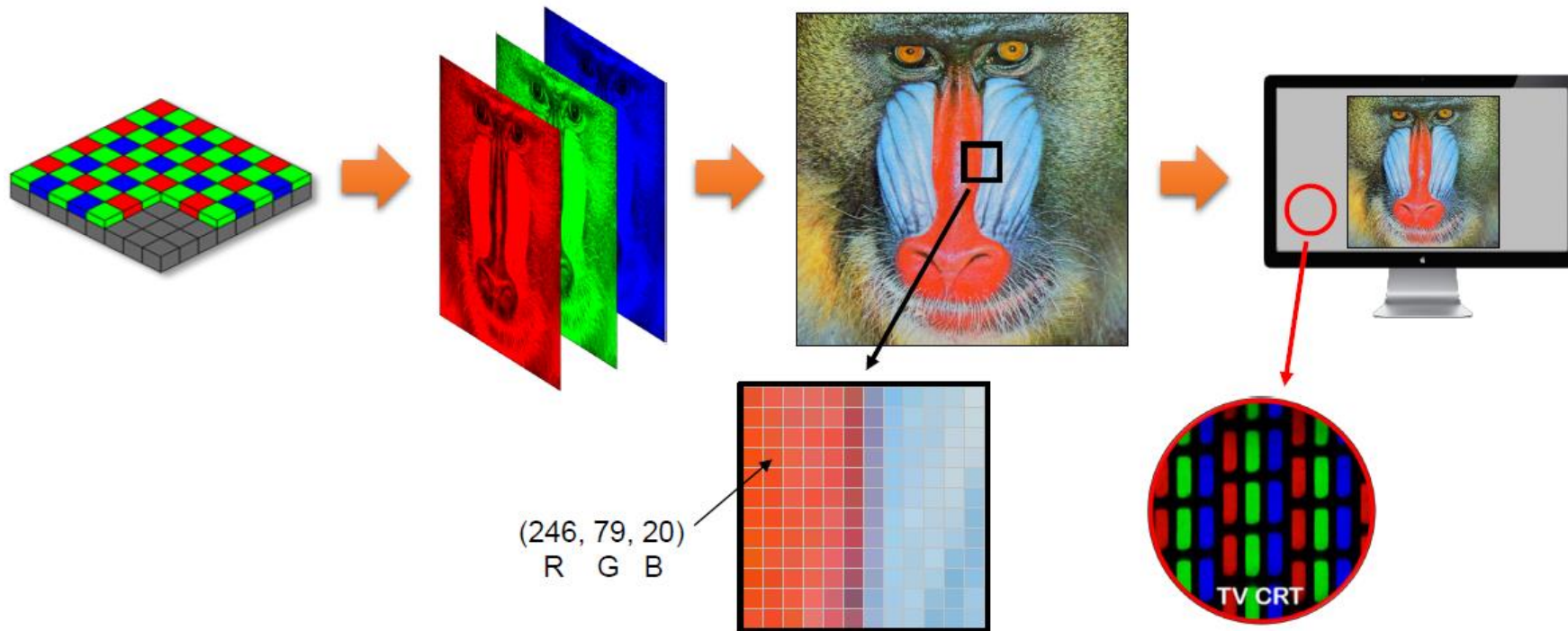


<https://youtu.be/wuhbqcMzOaw?t=7>

영상의 표현 방법

✔ 영상(image)이란?

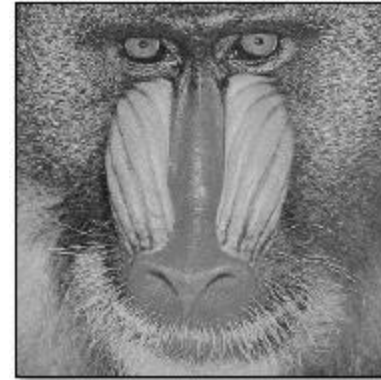
- 픽셀(pixel)이 바둑판 모양의 격자에 나열되어 있는 형태(2차원 행렬)
- 픽셀: 영상의 기본 단위, picture element, 화소(畵素)



영상의 표현 방법

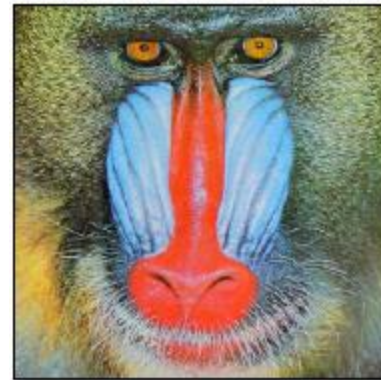
✓ 그레이스케일(grayscale) 영상

- 흑백 사진처럼 색상 정보가 없이 오직 밝기 정보만으로 구성된 영상
- 밝기 정보를 256 단계로 표현



✓ 트루컬러(truecolor) 영상

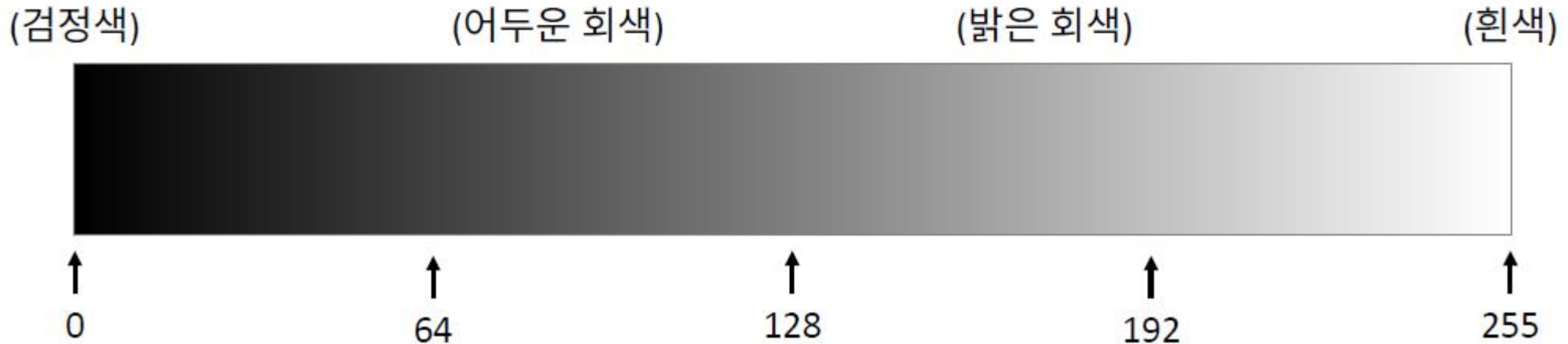
- 컬러 사진처럼 색상 정보를 가지고 있어서 다양한 색상을 표현할 수 있는 영상
- Red, Green, Blue 색 성분을 256 단계로 표현
→ $256^3 = 16,777,216$ 색상 표현 가능



영상의 표현 방법

✓ 그레이스케일 영상의 픽셀 값 표현

- 밝기 성분을 0 ~ 255 범위의 정수로 표현

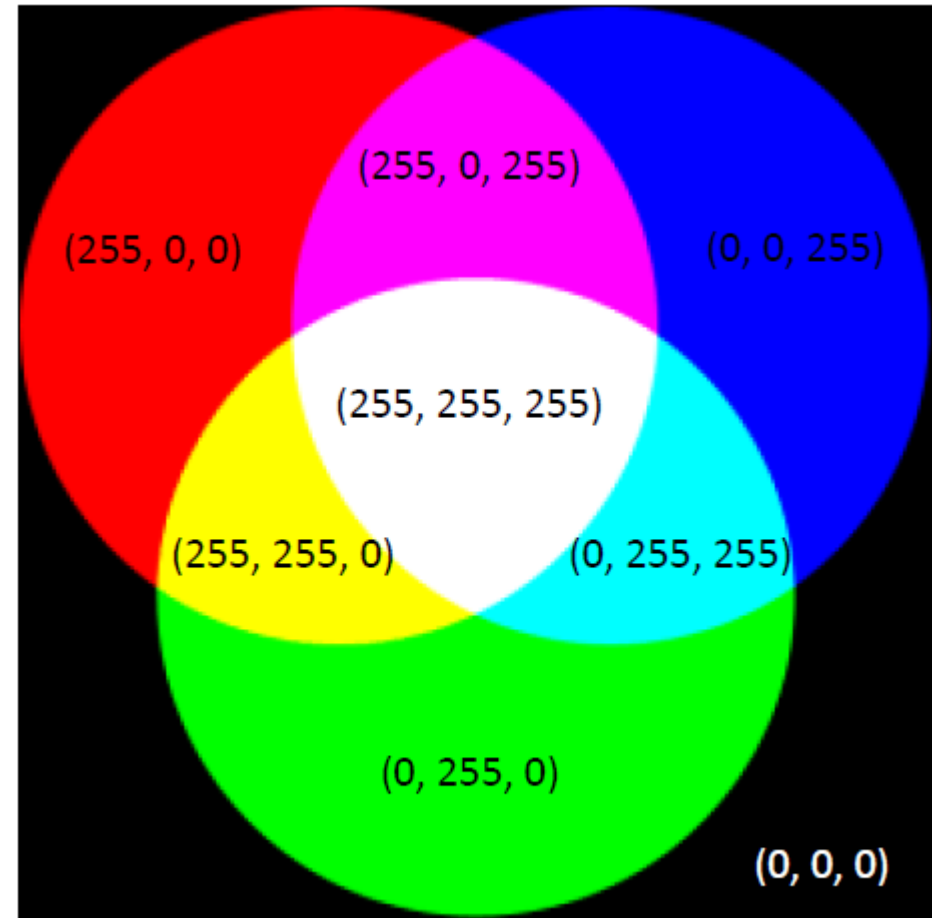


- 프로그래밍 언어에서 표현 방법: 1Byte 사용
 - C/C++ → unsigned char
 - Python → numpy.uint8

영상의 표현 방법

✓ 컬러 영상의 픽셀 값 표현

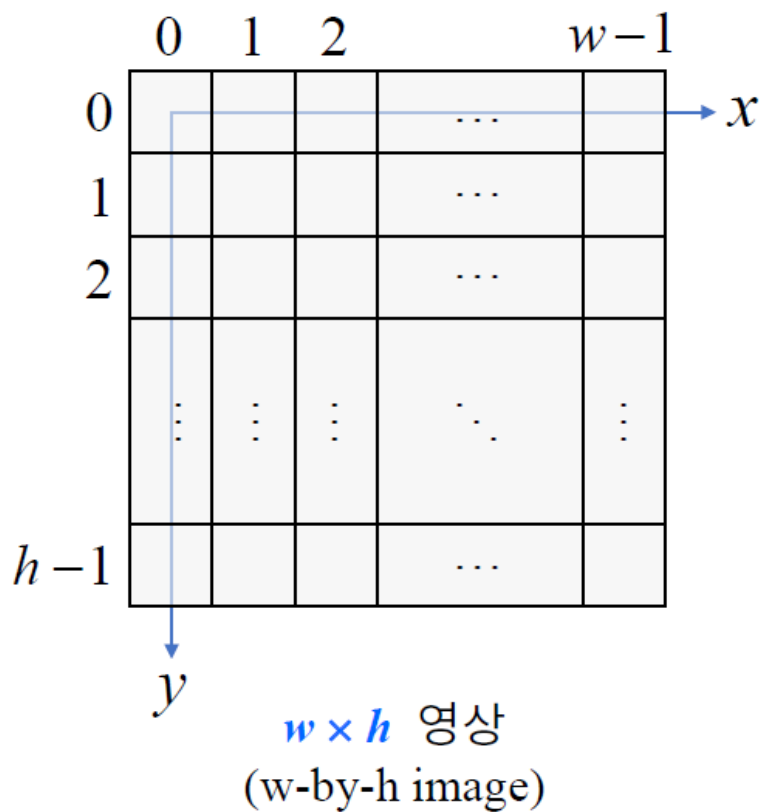
- R, G, B 색 성분의 크기를 각각 0 ~ 255 범위의 정수로 표현
 - 0: 해당 색성분이 전혀 없는 상태
 - 255: 해당 색 성분이 가득 있는 상태
- 프로그래밍 언어에서 표현 방법: 3Bytes 사용
 - C/C++ → 구조체, 클래스
 - Python → 튜플, numpy.ndarray



빛의 삼원색: (R, G, B)

영상의 표현 방법

✓ 영상에서 주로 사용되는 좌표계

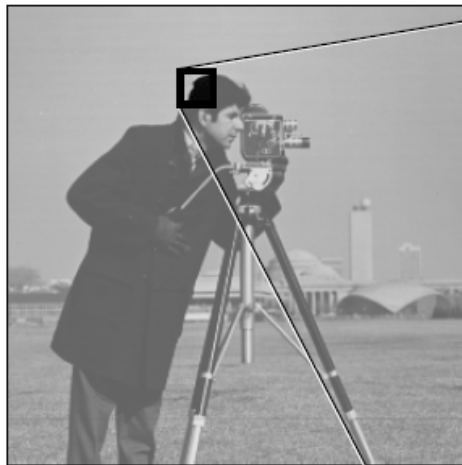


$$\mathbf{A} = \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,N} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ a_{M,1} & a_{M,2} & \cdots & a_{M,N} \end{bmatrix}$$

$M \times N$ 행렬
(m-by-n matrix)

영상의 표현 방법

✔ 그레이스케일 영상에서 픽셀 값 분포의 예

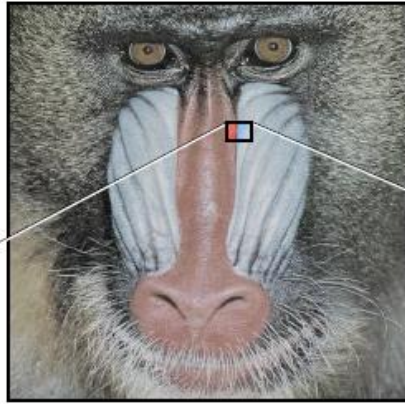


187	187	187	194	197	173	77	25	19	19
190	187	190	191	158	37	15	14	20	20
187	182	180	127	32	16	13	16	14	12
184	186	172	100	20	13	15	18	13	18
186	190	187	127	18	14	15	14	12	10
189	192	192	148	16	15	11	10	10	9
192	195	181	37	13	10	10	10	10	10
189	194	54	14	11	10	10	10	9	8
189	194	19	16	11	11	10	10	9	9
192	88	12	11	11	10	10	10	9	9

픽셀

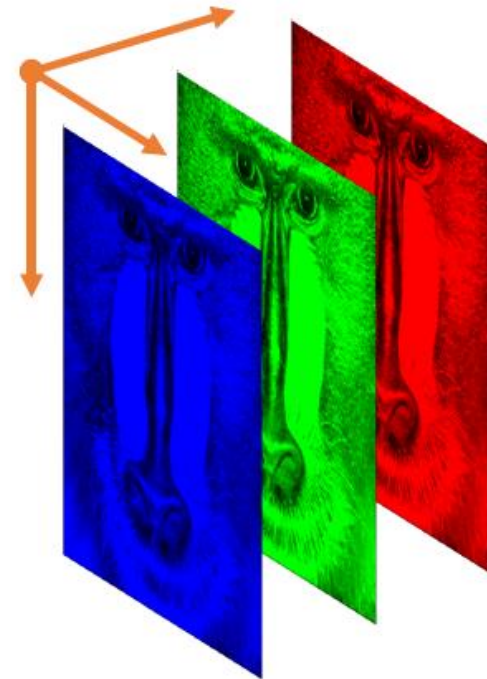
영상의 표현 방법

✔ 트루컬러 영상에서 픽셀 값 분포의 예



231	103	92	228	118	86	220	122	122	213	92	59	166	136	164	131	185	228	142	192	227	145	195	228
229	104	92	221	114	72	226	109	109	208	101	51	150	146	189	137	189	232	132	189	229	143	195	229
228	109	103	229	110	90	228	107	107	206	86	64	144	165	206	138	187	230	137	188	227	142	195	230
229	110	104	231	108	102	226	117	102	206	81	45	134	155	199	135	187	230	139	193	231	156	199	229
224	106	93	219	132	114	208	118	137	189	88	70	133	166	211	127	184	229	145	192	229	165	201	228
231	111	98	227	102	70	209	110	103	178	84	65	121	157	210	131	183	229	145	192	228	161	199	228

픽셀



영상 데이터의 크기

✓ 영상 데이터 크기 분석

- 그레이스케일 영상: (가로크기) × (세로크기) Bytes
- 트루컬러 영상: (가로크기) × (세로크기) × 3 Bytes



$$512 \times 512 = 262144 \text{ Bytes}$$



$$1920 \times 1080 \times 3 = 6220800 \text{ Bytes}$$
$$\approx 6 \text{ MBytes}$$

영상 파일 형식 특징

BMP

- 픽셀 데이터를 압축하지 않고 그대로 저장
→ 파일 용량이 큰 편
- 파일 구조가 단순해서 별도의 라이브러리 도움 없이 파일 입출력 프로그래밍 가능

JPG

- 주로 사진과 같은 컬러 영상을 저장
- 손실 압축(lossy compression)
- 압축률이 좋아서 파일 용량이 크게 감소
→ 디지털 카메라 사진 포맷으로 주로 사용

GIF

- 256 색상 이하의 영상을 저장
→ 일반 사진을 저장 시 화질 열화가 심함
- 무손실 압축(lossless compression)
- 움직이는 GIF 지원

PNG

- Portable Network Graphics
- 무손실 압축 (컬러 영상도 무손실 압축)
- 알파 채널(투명도)을 지원

OpenCV 개요

✓ What is OpenCV?

- Open source
- Computer vision & machine learning
- Software library

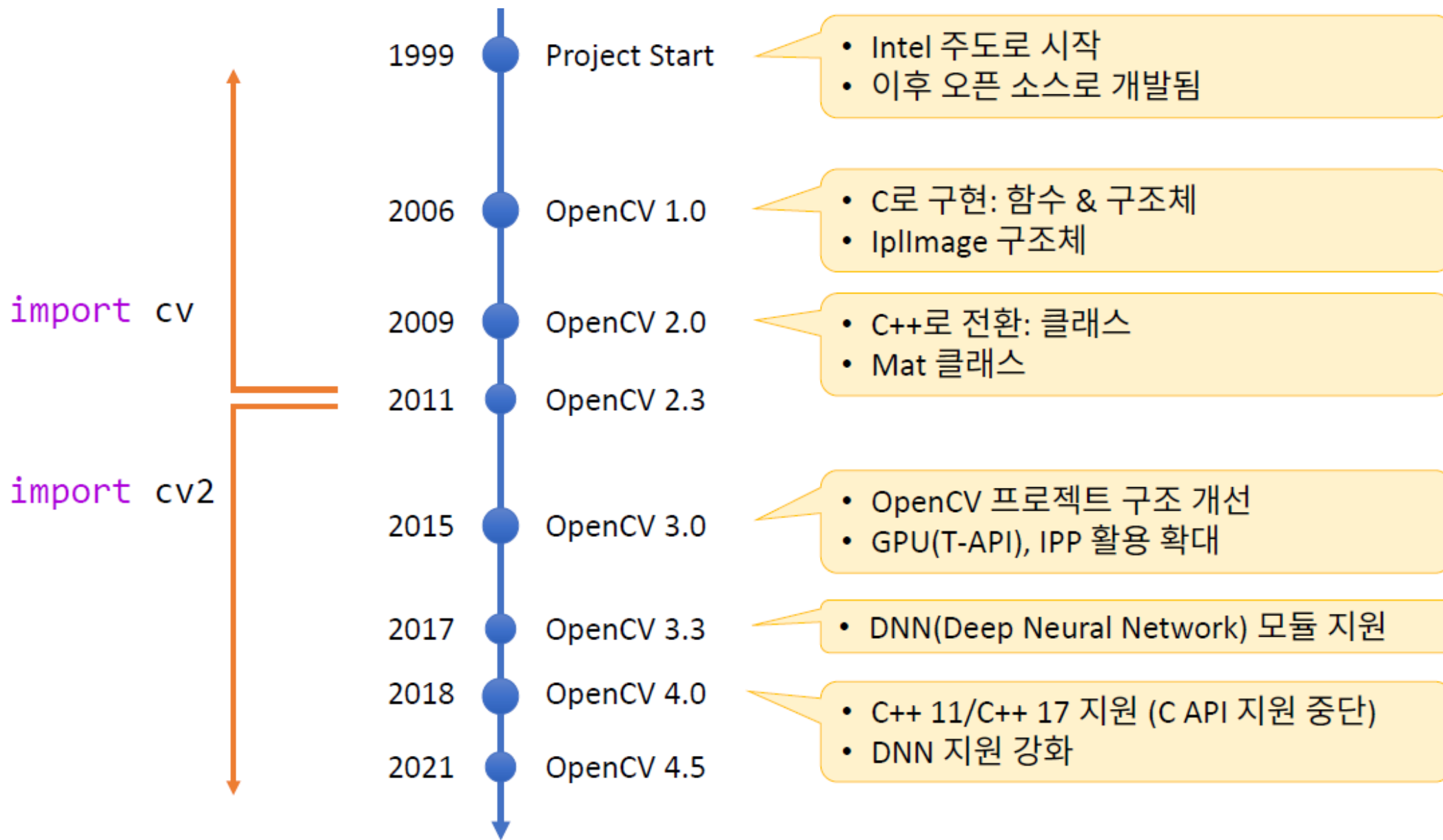
✓ Why OpenCV?

- BSD/Apache 2 license ... Free for academic & commercial use
- Multiple interface ... C, C++, Python, Java, JavaScript, MATLAB, etc.
- Multiple platform ... Windows, Linux, Mac OS, iOS, Android
- Optimized ... CPU instructions, Multi-core processing, OpenCL, CUDA
- Popular ... More than 18 million downloads



<http://www.opencv.org/>

OpenCV 역사

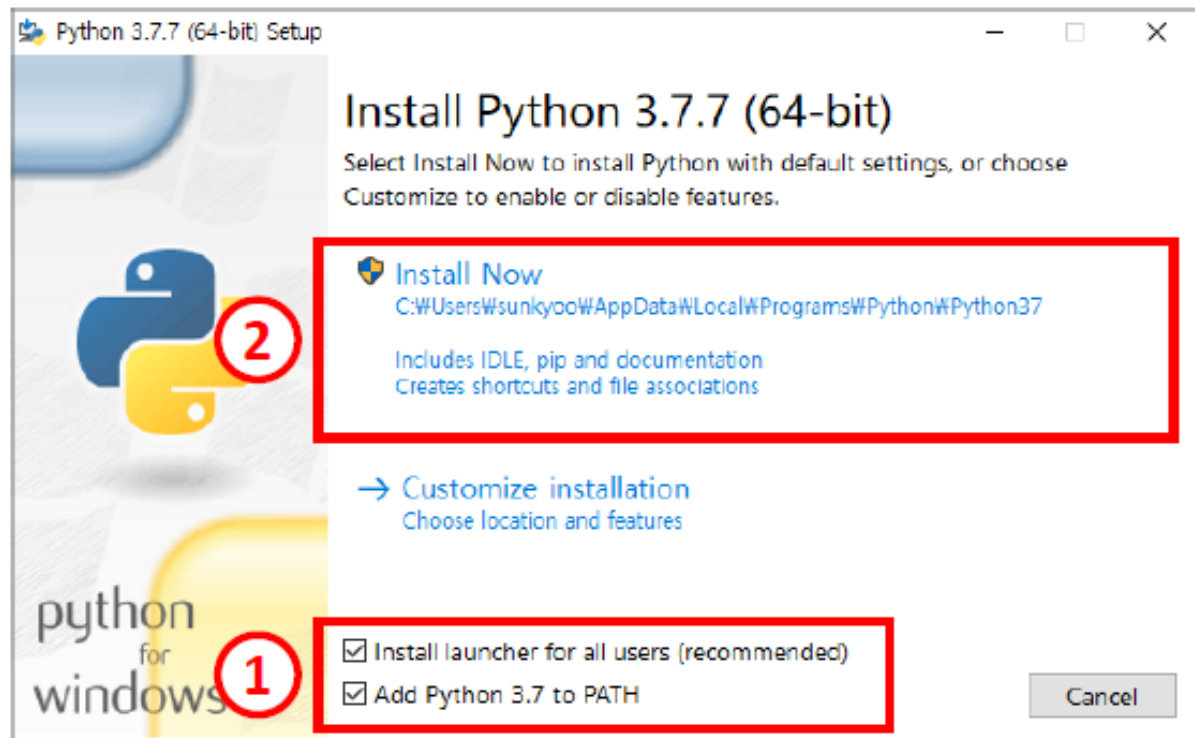


Python 설치

✓ Python 설치하기



<https://www.python.org/downloads/>



Python 설치

✓ 파이참 설치하기

JETBRAINS Developer Tools Team Tools Education Solutions Support Store

PyCharm New UI What's New Features Learn Pricing **Download** 1

Windows macOS Linux

PyCharm Professional
The Python IDE for Professional Developers

Download .exe
Free 30-day trial

Version: 2023.2 Build: 232.8660.197 26 July 2023
[System requirements](#) [Other versions](#)
[Installation](#) [Instructions](#) [Third-party software](#)

We value the vibrant Python community, and that's why we proudly offer the PyCharm Community Edition for free, as our open-source contribution to support the Python ecosystem.

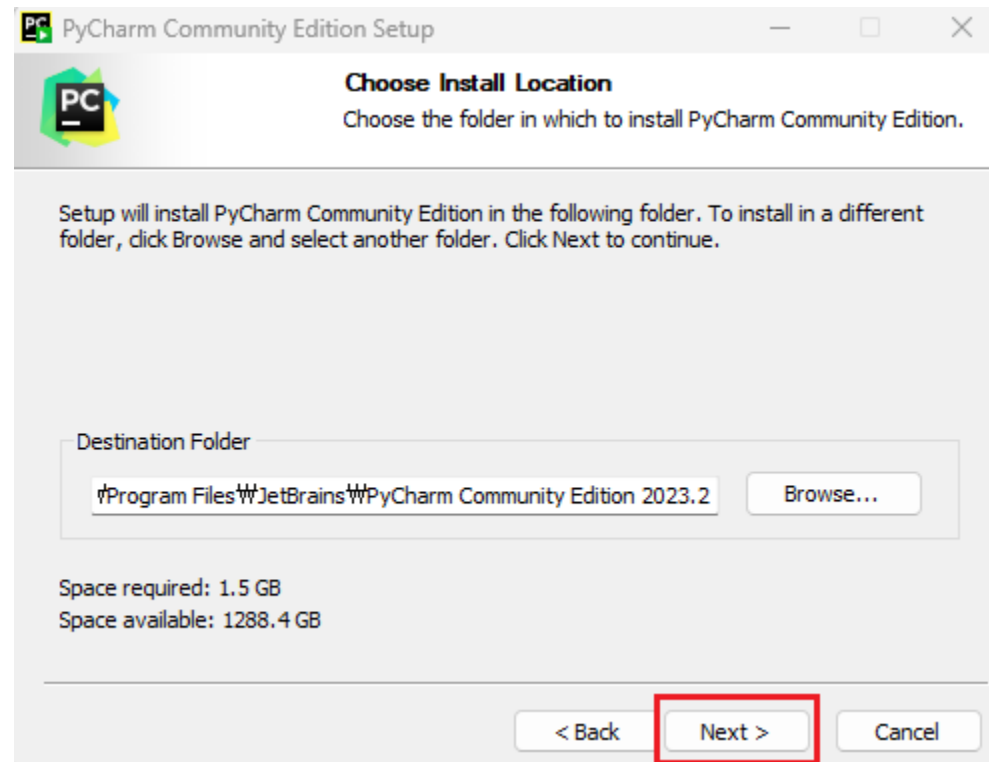
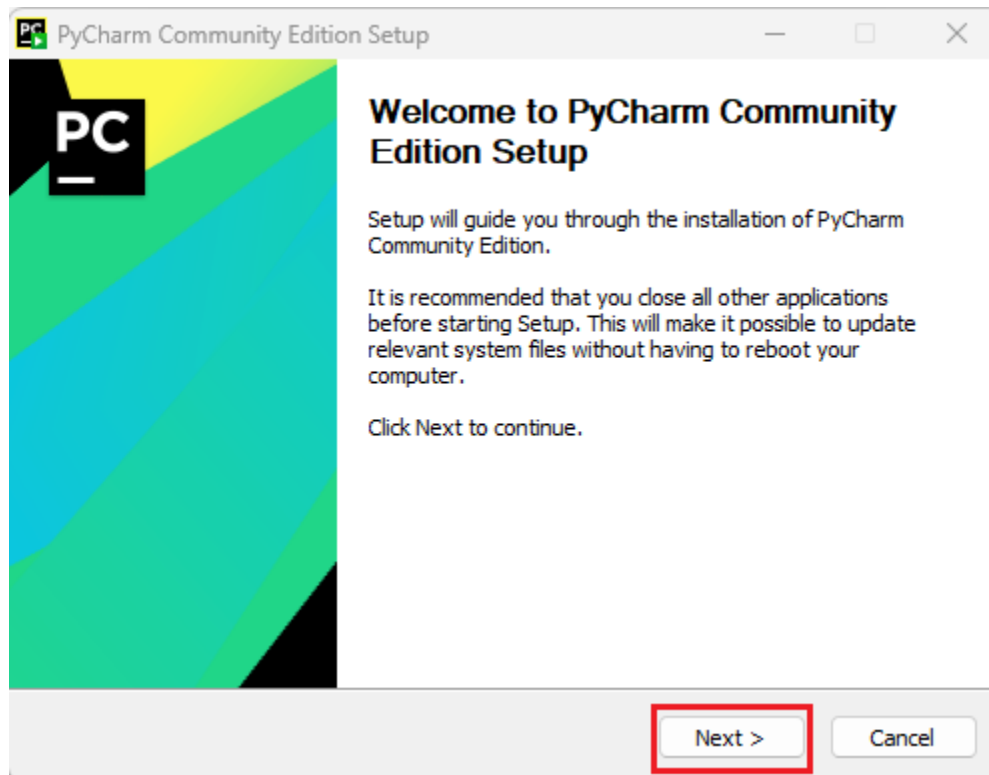
PyCharm Community Edition
The IDE for Pure Python Development

Download .exe 2
Free, built on open source

<https://www.jetbrains.com/pycharm/download/?section=windows#section=windows>

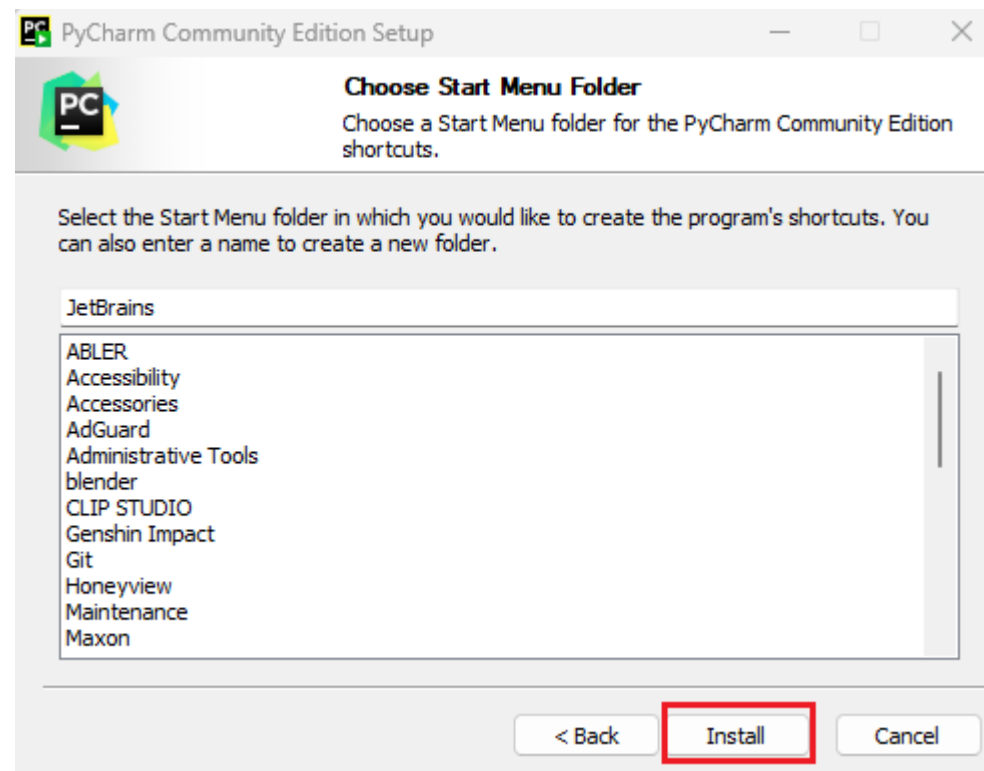
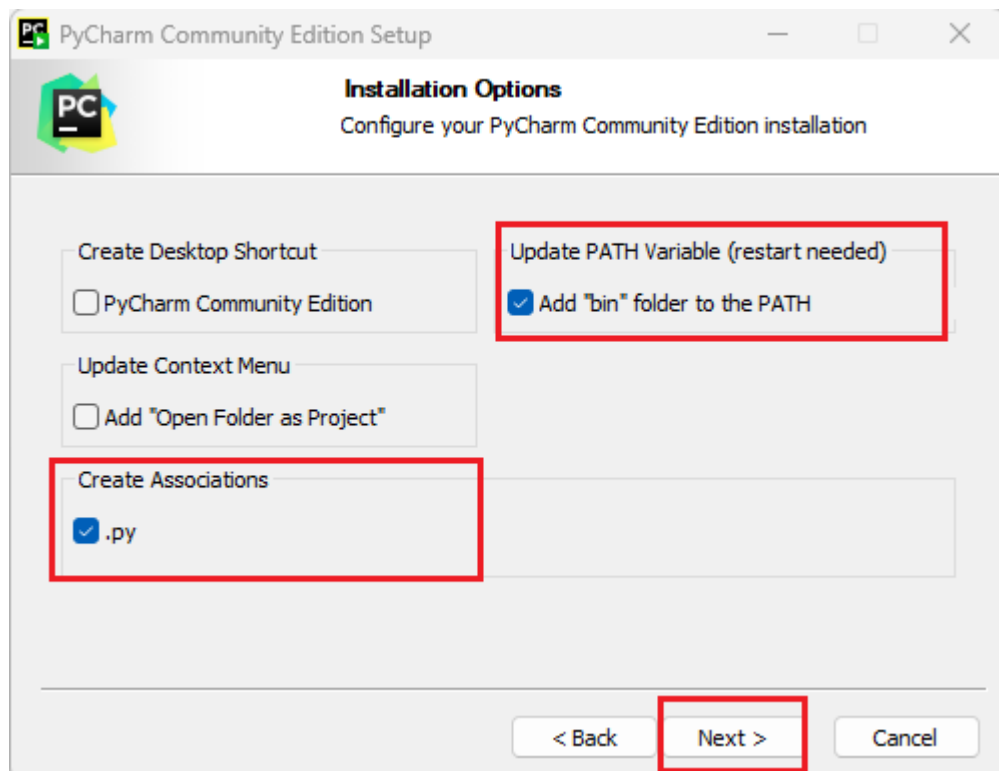
Python 설치

✓ 파이참 설치하기



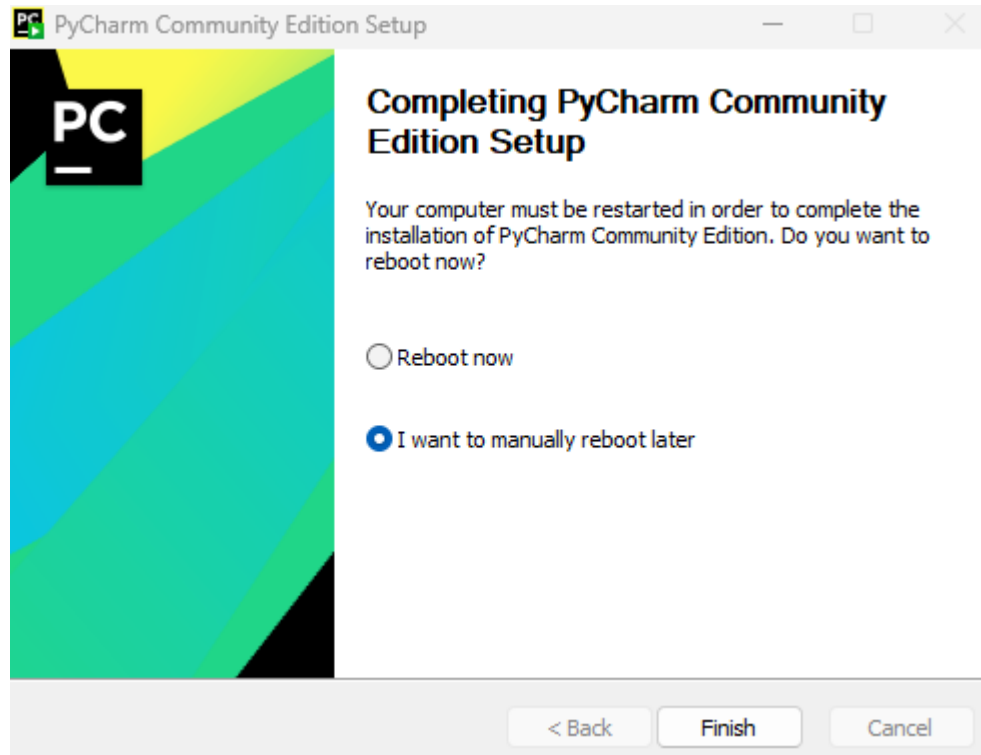
Python 설치

✓ 파이참 설치하기



Python 설치

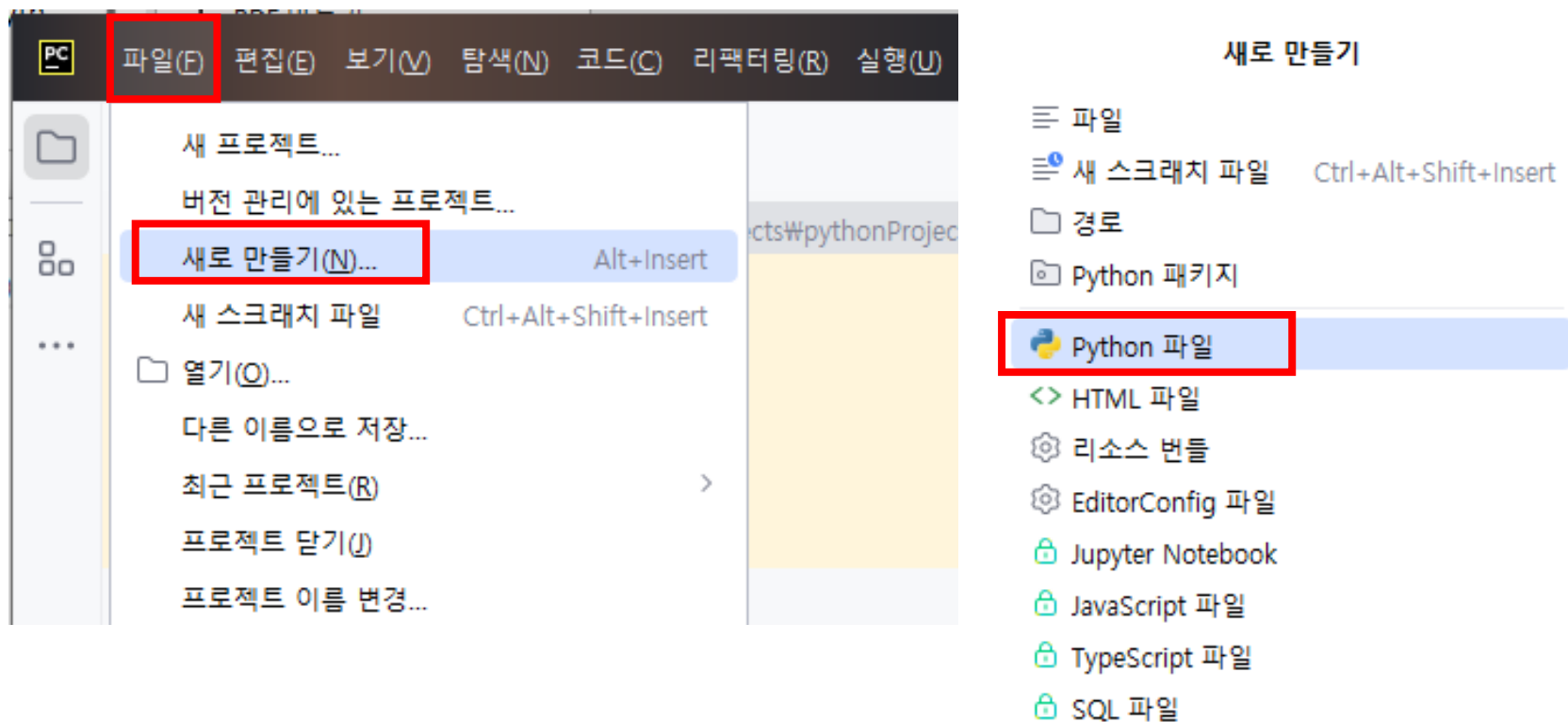
✓ 파이참 설치하기



HelloCV.py 프로그램 만들기

✓ 파이참에서 새 Python 프로그램 만들기

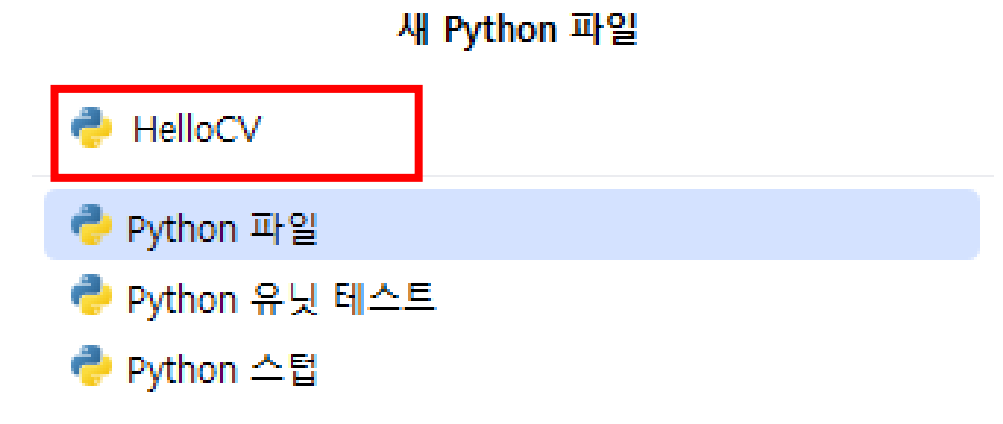
- 아래 그림처럼 [새로 만들기] 메뉴 클릭 후, Python 파일 선택



HelloCV.py 프로그램 만들기

✓ 파이참에서 새 Python 프로그램 만들기

- 아래 그림처럼 HelloCV 파일 이름 입력



HelloCV.py 프로그램 만들기

✓ 파이참에서 새 Python 프로그램 만들기

- 편집창에 소스코드 입력

```
import sys
import cv2
```

```
print('Hello OpenCV', cv2.__version__)
```

```
img = cv2.imread('cat.bmp')
```

← cat.bmp 파일을 불러와 img 변수에 저장

```
if img is None:
    print('Image load failed!')
    sys.exit()
```

← 영상 파일 불러오기가 실패하면 에러 메시지를 출력하고 종료.

```
cv2.namedWindow('image')
cv2.imshow('image', img)
cv2.waitKey()
```

← "image"라는 이름의 새 창을 만들고,
이 창에 img 영상을 출력하고,
키보드 입력이 있을 때까지 대기.

```
cv2.destroyAllWindows()
```

← 생성된 모든 창을 닫음

OpenCV API

✓ 영상 파일 불러오기

cv2.imread(filename, flags=None) -> retval

- filename: 불러올 영상 파일 이름(문자열)
- flags: 영상 파일 불러오기 옵션 플래그

cv2.IMREAD_COLOR	BGR 컬러 영상으로 읽기 (기본값) shape = (rows, cols, 3)
cv2.IMREAD_GRAYSCALE	그레이스케일 영상으로 읽기 shape = (rows, cols)
cv2.IMREAD_UNCHANGED	영상 파일 속성 그대로 읽기 (e.g.) 투명한 PNG 파일: shape = (rows, cols, 4)

- retval: 불러온 영상 데이터 (numpy.ndarray)

✓ 영상 파일 저장하기

```
cv2.imwrite(filename, img, params=None) -> retval
```

- filename: 저장할 영상 파일 이름 (문자열)
- img: 저장할 영상 데이터(numpy.ndarray)
- params: 파일 저장 옵션 지정 (속성& 값의 정수 쌍)
e.g) [cv2.IMWRITE_JPEG_QUALITY, 90] : JPG 파일 압축률을 90%로 지정
- retval: 정상적으로 저장하면 True, 실패하면 False.

OpenCV API

✓ 새 창 띄우기

cv2.namedWindow(winname, flags=None) -> None

- winname: 창 고유 이름(문자열)
- flags: 창 속성 지정 플래그

cv2.WINDOW_NORMAL	영상 크기를 창 크기에 맞게 지정
cv2.WINDOW_AUTOSIZE	창 크기를 영상 크기에 맞게 변경 (기본값)

✓ 창 닫기

```
cv2.destroyWindow(winname) -> None  
cv2.destroyAllWindows() -> None
```

- winname: 닫고자 하는 창 이름
- 참고사항
 - cv2.destroyWindow() 함수는 지정한 창 하나만 닫고, cv2.destroyAllWindows() 함수는 열려 있는 모든 창을 닫음.
 - 일반적인 경우 프로그램 종료 시 운영체제에 의해 열려 있는 모든 창이 자동으로 닫힘

✓ 창 위치 이동

```
cv2.moveWindow(winname, x, y) -> None
```

- winname: 창 이름
- x, y: 이동할 위치 좌표

✓ 창 크기 변경

```
cv2.resizeWindow(winname, width, height) -> None
```

- winname: 창 이름
- width: 변경할 창의 가로 크기
- height: 변경할 창의 세로 크기
- 참고사항
 - 창 생성시 cv2.WINDOW_NORMAL 속성으로 생성되어야 동작
 - 영상 출력 부분의 크기만을 고려함(제목 표시줄, 창 경계는 고려되지 않음)

✓ 영상 출력하기

```
cv2.imshow(winname, mat) -> None
```

- winname: 영상을 출력할 대상 창 이름
- mat: 출력할 영상 데이터(numpy.ndarray)
- 참고사항
 - uint16, int32 자료형 행렬의 경우, 행렬 원소 값을 255로 나눠서 출력
 - float32, float64 자료형 행렬의 경우, 행렬 원소 값에 255를 곱해서 출력
 - 만약 winname에 해당하는 창이 없으면 창을 새로 만들어서 영상을 출력함
 - Windows 운영체제에서는 Ctrl + C (복사), Ctrl + S (저장) 지원
 - 실제로는 cv2.waitKey() 함수를 호출해야 화면에 영상이 나타남

✓ 키보드 입력 대기

cv2.waitKey(delay=None) -> retval

- delay: 밀리초 단위 대기 시간. $\text{delay} \leq 0$ 이면 무한히 기다림. 기본값은 0.
- retval: 눌린 키 값(ASCII code). 키가 눌리지 않으면 -1.
- 참고사항
 - cv2.waitKey() 함수는 OpenCV 창이 하나라도 있을 때 동작함
 - 특정 키 입력을 확인하려면 ord() 함수를 이용

```
while True:
    if cv2.waitKey() == ord('q'):
        break
```

- 주요 특수키 코드: 27(ESC), 13(ENTER), 9(TAB)

Matplotlib을 이용한 영상 출력

✓ Matplotlib 라이브러리

- 함수 그래프, 차트(chart), 히스토그램(histogram) 등의 다양한 그리기 기능을 제공하는 Python 패키지

✓ 컬러 영상 출력

- 컬러 영상의 색상 정보가 RGB 순서이어야 함
- cv2.imread() 함수로 불러온 영상의 색상 정보는 BGR 순서이므로 이를 RGB 순서로 변경해야 함
→ cv2.cvtColor() 함수 이용

✓ 그레이스케일 영상 출력

- plt.imshow() 함수에서 컬러맵을 cmap='gray ' 으로 지정

Matplotlib을 이용한 영상 출력

✓ Matplotlib을 이용하여 영상 출력하기

```
import matplotlib.pyplot as plt
import cv2

# 컬러 영상 출력
imgBGR = cv2.imread('cat.bmp')
imgRGB = cv2.cvtColor(imgBGR, cv2.COLOR_BGR2RGB)

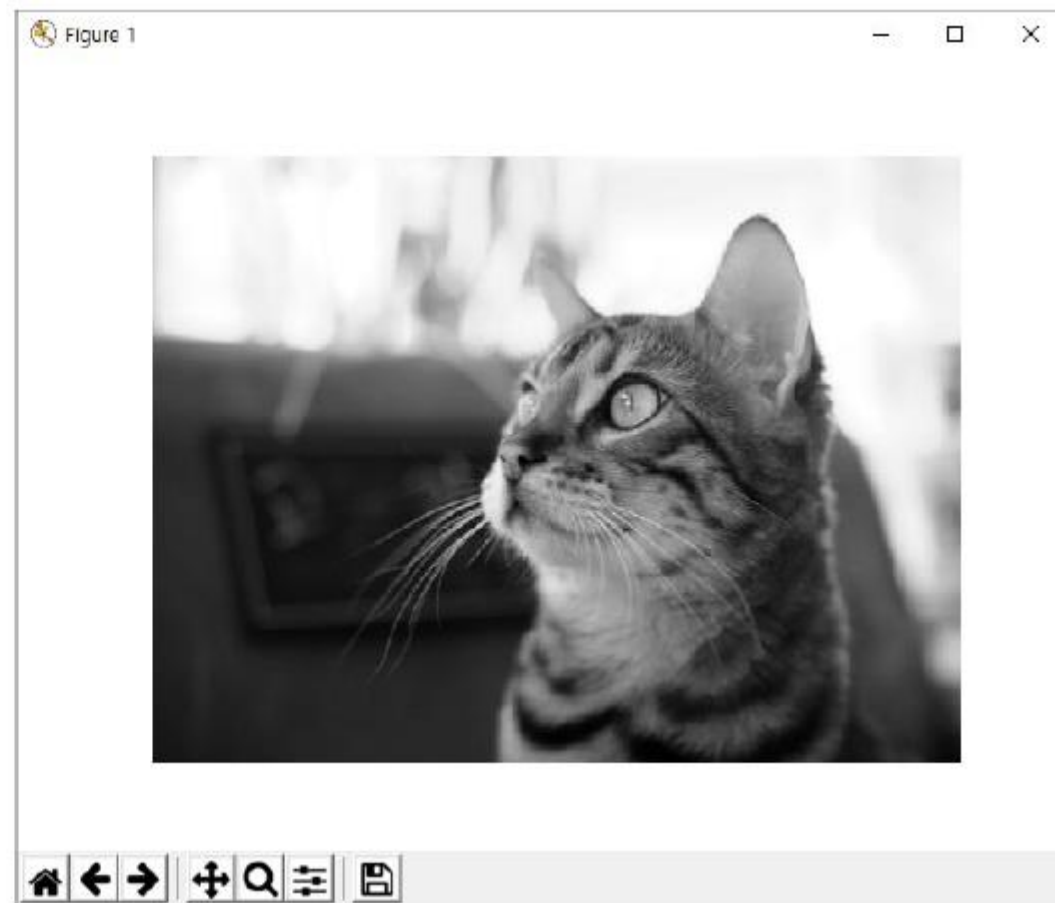
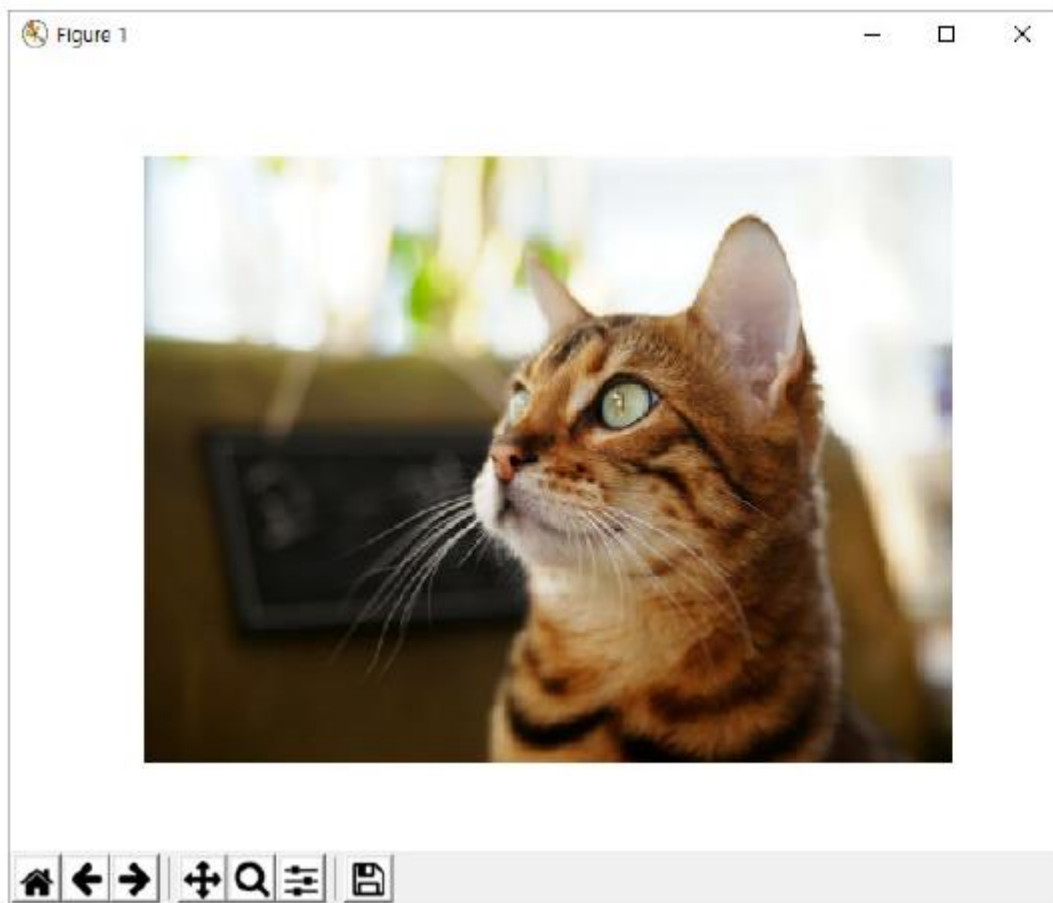
plt.axis('off')
plt.imshow(imgRGB)
plt.show()

# 그레이스케일 영상 출력
imgGray = cv2.imread('cat.bmp', cv2.IMREAD_GRAYSCALE)

plt.axis('off')
plt.imshow(imgGray, cmap='gray')
plt.show()
```


Matplotlib을 이용한 영상 출력

✔ Matplotlib을 이용하여 영상 출력하기



Matplotlib을 이용한 영상 출력

✓ Matplotlib을 이용하여 창 하나에 여러 개의 이미지 출력하기

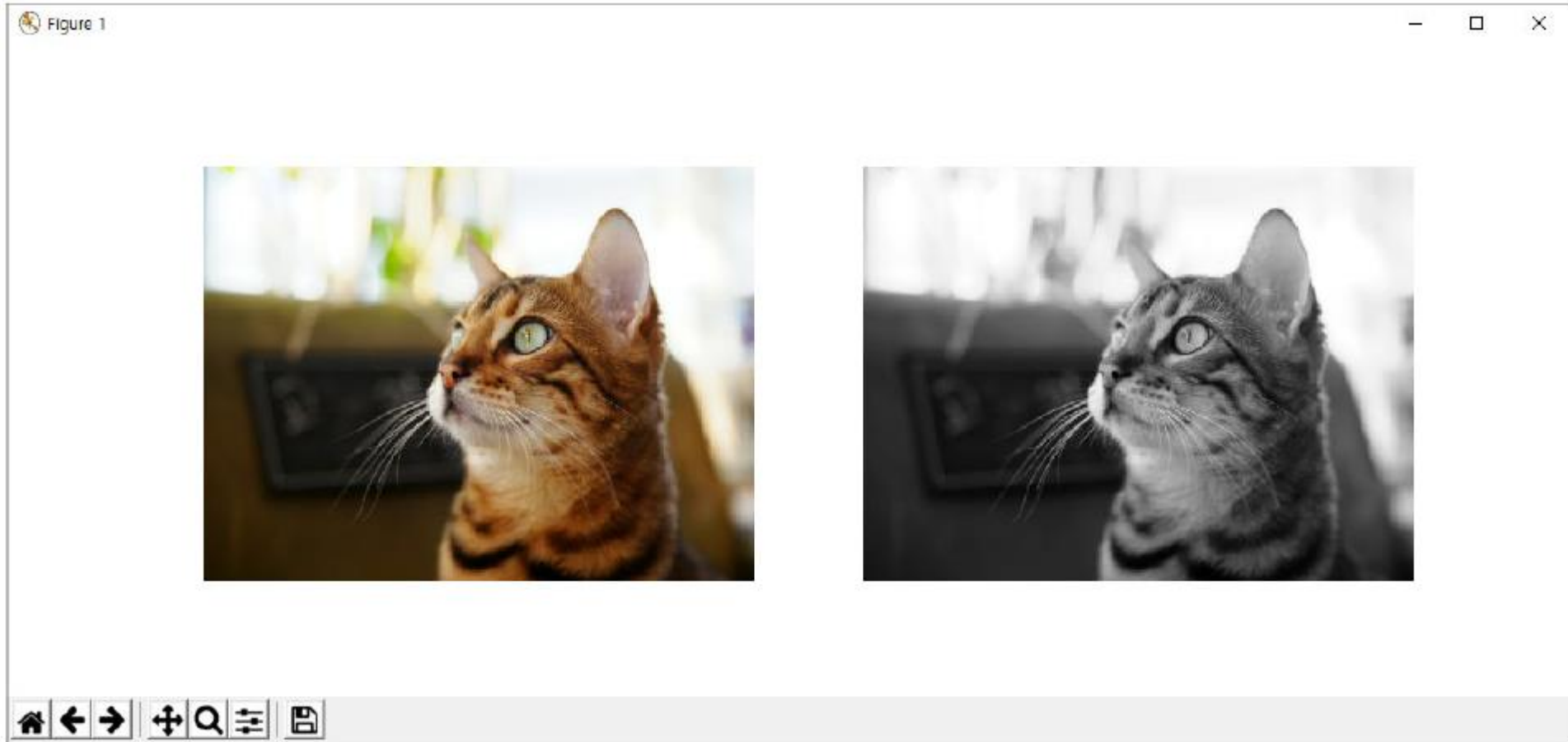
```
import matplotlib.pyplot as plt
import cv2

# 컬러 영상 & 그레이스케일 영상 불러오기
imgBGR = cv2.imread('cat.bmp')
imgRGB = cv2.cvtColor(imgBGR, cv2.COLOR_BGR2RGB)
imgGray = cv2.imread('cat.bmp', cv2.IMREAD_GRAYSCALE)

# 두 개의 영상을 함께 출력
plt.subplot(121), plt.axis('off'), plt.imshow(imgRGB)
plt.subplot(122), plt.axis('off'), plt.imshow(imgGray, cmap='gray')
plt.show()
```

Matplotlib을 이용한 영상 출력

✔ Matplotlib을 이용하여 창 하나에 여러 개의 이미지 출력하기



[프로젝트] 이미지 슬라이드쇼

✓ 이미지 슬라이드쇼

- 특정 폴더에 있는 모든 이미지 파일을 이용하여 슬라이드쇼를 수행

✓ 구현할 기능

- 특정 폴더에 있는 이미지 파일 목록 읽기
- 이미지를 전체 화면으로 출력하기
- 일정 시간동안 이미지를 화면에 출력하고, 다음 이미지로 교체하기(무한루프)

[프로젝트] 이미지 슬라이드쇼

✓ 특정 폴더에 있는 이미지 파일(*.jpg) 목록 읽기

- os.listdir()

```
import os

file_list = os.listdir('.\\images')
img_files = [os.path.join('.\\images', file)
              for file in file_list if file.endswith('.jpg')]
```

- glob.glob()

```
import glob

img_files = glob.glob('.\\images\\*.jpg')
```


[프로젝트] 이미지 슬라이드쇼

✓ 전체 화면 영상 출력창 만들기

- 먼저 cv2.WINDOW_NORMAL 속성의 창을 만든 후,
cv2.setWindowProperty() 함수를 사용하여 전체 화면 속성으로 변경

```
cv2.namedWindow('image', cv2.WINDOW_NORMAL)  
cv2.setWindowProperty('image', cv2.WND_PROP_FULLSCREEN,  
                        cv2.WINDOW_FULLSCREEN)
```

[프로젝트] 이미지 슬라이드쇼

✓ 불러온 영상을 반복적으로 출력하기

```
cnt = len(img_files)
idx = 0

while True:
    img = cv2.imread(img_files[idx])

    if img is None:
        print('Image load failed!')
        break

    cv2.imshow('image', img)
    if cv2.waitKey(1000) >= 0:
        break

    idx += 1
    if idx >= cnt:
        idx = 0
```

[프로젝트] 이미지 슬라이드쇼

✓ 전체 코드

```
1 import os
2 import sys
3 import glob
4 import cv2
5
6
7 # 이미지 파일을 모두 img_files 리스트에 추가
8
9 # os.listdir() 사용 방법
10 #file_list = os.listdir('.\\images')
11 #img_files = [os.path.join('.\\images', file) for file in file_list if file.endswith('.jpg')]
12
13 # glob.glob() 사용 방법
14 img_files = glob.glob('.\\images\\*.jpg')
15
16 if not img_files:
17     print("There are no jpg files in 'images' folder")
18     sys.exit()
19
20 # 전체 화면으로 'image' 창 생성
21 cv2.namedWindow( winname: 'image', cv2.WINDOW_NORMAL)
22 cv2.setWindowProperty( winname: 'image', cv2.WND_PROP_FULLSCREEN, cv2.WINDOW_FULLSCREEN)
23
```

⚠ 1 ⚠ 2 ✓ 2 ^

[프로젝트] 이미지 슬라이드쇼

✓ 전체 코드

```
24     # 무한 루프
25     cnt = len(img_files)
26     idx = 0
27
28     while True:
29         img = cv2.imread(img_files[idx])
30
31         if img is None:
32             print('Image load failed!')
33             break
34
35         cv2.imshow(winname='image', img)
36         if cv2.waitKey(1000) >= 0:
37             break
38
39         idx += 1
40         if idx >= cnt:
41             idx = 0
42
43     cv2.destroyAllWindows()
```