

## OpenCV: 이미지 reading and writing

# 1. OpenCV 이미지 설치

- 1.1 테스트 환경

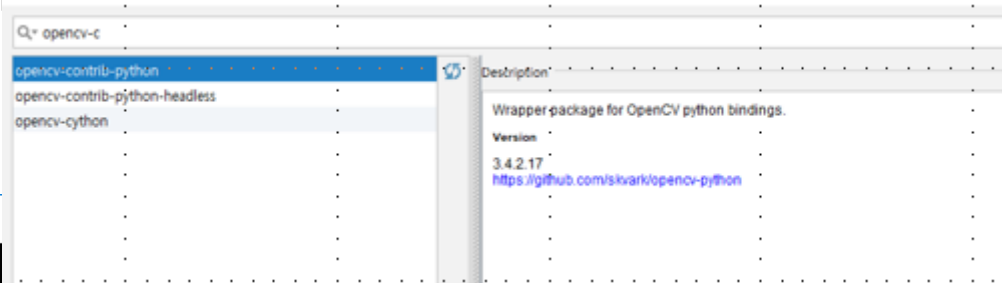
순 번	제 목	내 용
1	운영체제	윈도우 10
2	프로그래밍언어	Python

# 1. OpenCV 이미지 설치

## 1.1 테스트 환경

opencv-python 3.4.2.17

pip install opencv-python



명령 프롬프트

```
Microsoft Windows [Version 10.0.17134.165]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\User>cd C:\Users\User\AppData\Local\Programs\Python\Python36\Lib\site-packages

C:\Users\User\AppData\Local\Programs\Python\Python36\Lib\site-packages>
C:\Users\User\AppData\Local\Programs\Python\Python36\Lib\site-packages>
C:\Users\User\AppData\Local\Programs\Python\Python36\Lib\site-packages>
C:\Users\User\AppData\Local\Programs\Python\Python36\Lib\site-packages>
C:\Users\User\AppData\Local\Programs\Python\Python36\Lib\site-packages>pip install opencv_python-3.4.2.17-cp36-cp36m-win_
amd64.whl
Processing c:\Users\User\AppData\Local\Programs\Python\Python36\Lib\site-packages\opencv_python-3.4.2.17-cp36-cp36m-win_
amd64.whl
Requirement already satisfied: numpy>=1.11.3 in c:\Users\User\AppData\Local\Programs\Python\Python36\Lib\site-packages (
from opencv-python==3.4.2.17)
Installing collected packages: opencv-python
Successfully installed opencv-python-3.4.2.17
You are using pip version 9.0.1, however version 18.0 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.

C:\Users\User\AppData\Local\Programs\Python\Python36\Lib\site-packages>
```

# 1. OpenCV 이미지 예제

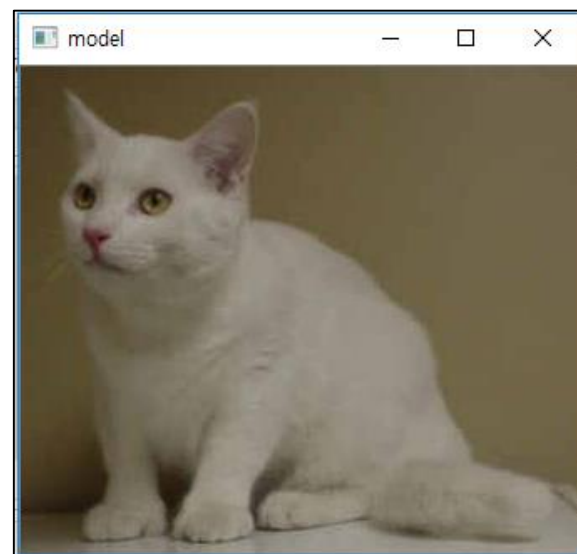
## ▪ 1.1 예제 1 - 이미지 띄우기

```
import numpy as np
import cv2

def showImage():
    imgfile = 'images/cat.jpg'
    img = cv2.imread(imgfile, cv2.IMREAD_COLOR)

    cv2.namedWindow('cat', cv2.WINDOW_NORMAL)
    cv2.imshow('cat', img)
    cv2.waitKey(0)
    cv2.destroyAllWindows()

showImage()
```



[그림] 예제 1 실행 결과

# OpenCV 이미지 예제

## ▪ 1.1 예제 1 - 이미지 띄우기

```
>> img = cv2.imread(imgfile, cv2.IMREAD_COLOR)
```

- 이미지를 읽고 화면에 디스플레이 하기 위한 함수를 만들어본다. opencv의 imread 함수는 이미지 파일을 읽기 위한 객체를 리턴한다. 이 함수의 첫번째 인자는 읽고자 하는 이미지 파일경로, 두번째 인자는 이미지 파일을 읽는 방식을 나타내는 플래그이다.

# OpenCV 이미지 예제

- 1.1 예제 1 - 이미지 띄우기

```
>> img = cv2.imread(imgfile, cv2.IMREAD_COLOR)
```

- 이미지 읽기 플래그는 총 3가지가 있다.

순 번	제 목	내 용
1	cv2.IMREAD_COLOR	컬러 이미지 형태로 불러온다. 이미지의 투명한 부분은 모두 무시된다. 정수 값으로는 1이다.
2	cv2.IMREAD_GRAYSCALE	흑백 이미지 형태로 불러온다. 정수 값으로는 0이다.
3	cv2.IMREAD_UNCHANGED	알파채널을 포함하여 이미지를 그대로 불러온다. 정수 값으로는 -1이다.

# OpenCV 이미지 예제

- 1.1 예제 1 - 이미지 띄우기

```
>> cv2.namedWindow('model', cv2.WINDOW_NORMAL)
```

- 이미지 크기를 변경시킬 수 있는 창을 만들기 위해서 cv2.namedWindow() 함수를 이용해 이미지가 표시될 창의 성격을 지정한다. 이 함수의 첫번째 인자는 생성될 이미지 윈도우의 타이틀, 두 번째 인자는 윈도우 사이즈 플래그 이다. 플래그는 2가지가 있다.

순 번	제 목	내 용
1	cv2.WINDOW_AUTOSIZE	원본 이미지 크기로 고정하여 윈도우 생성
2	cv2.WINDOW_NORMAL	원본 이미지 크기로 윈도우를 생성하여 이미지를 나타내지만 사용자가 마우스를 이용하여 크기를 조정할 수 있다.

# OpenCV 이미지 예제

## ▪ 1.1 예제 1 - 이미지 띄우기

```
>> cv2.imshow('cat', img)
```

- `cv2.imread()`에 의해 반환된 이미지 객체 `img`를 화면에 나타내기 위한 함수이다. 이 함수의 첫 번째 인자는 윈도우 타이틀 이며, 두 번째 인자는 화면에 표시할 이미지 객체이다.

```
>> cv2.waitKey(0)
```

- 화면에 이미지를 표시한 후 사용자가 키보드를 누를 때 까지 대기하라는 함수이다.  
`cv2.waitKey()` 함수는 지정된 시간동안 키보드 입력을 기다린다. `cv2.waitKey(5)`는 5ms 동안 대기하라는 의미이다. `cv2.waitKey(0)`은 키보드 입력이 있을 때 까지 기다리라는 의미이다.  
이 함수의 리턴 값은 사용자가 누른 키보드의 값이다.



# OpenCV 이미지 예제

- 1.1 예제 1 - 이미지 띄우기

```
>> cv2.destroyAllWindows()
```

- 생성한 모든 윈도우를 제거한다.
- 이렇게 생성한 `showImage` 함수를 실행하면 지정한 사진이 화면에 나타난다. 아무키나 눌러 프로그램을 종료한다.

# OpenCV 이미지 예제

## ▪ 1.2 예제 2 - 이미지 파일 복사

```
import numpy as np
import cv2

def showImage():
    imgfile = 'images/cat.jpg'
    img = cv2.imread(imgfile, cv2.IMREAD_COLOR)
    cv2.imshow('cat', img)

    k = cv2.waitKey(0) & 0xFF

    if k == 27:
        cv2.destroyAllWindows()
    elif k == ord('c'):
        cv2.imwrite('images/cat_copy.jpg', img)
        cv2.destroyAllWindows()

showImage()
```



[그림] 예제 2 실행 결과

# OpenCV 이미지 예제

- 1.2 예제 2 - 이미지 파일 복사

```
>>k = cv2.waitKey(0) & 0xFF
```

- 사용자가 키보드로 입력할 때까지 기다리며 입력이 되면 그 값을 k로한다. 64비트 컴퓨터에서는 0xFF와 연산해주는데 이는 byte형에서 int형으로 형변환 할 때 0이 1로 변하는 에러를 잡아준다.

# OpenCV 이미지 예제

- 1.2 예제 2 - 이미지 파일 복사

```
>> if k == 27:
```

```
    cv2.destroyAllWindows()
```

```
elif k == ord('c'):
```

```
    cv2.imwrite('images/cat_copy.jpg', img)
```

```
    cv2.destroyAllWindows()
```

- 아스키 코드값이 27인 ESC키를 누르면 프로그램을 종료한다. c키를 누르면 images 폴더에 cat\_copy.jpg라는 이름의 복사본을 만든다. 여기서 ord()함수는 문자를 아스키 값으로 변환하는 함수이다. 복사본이 폴더에 생성된 후 프로그램이 종료된다.

# OpenCV 이미지 예제

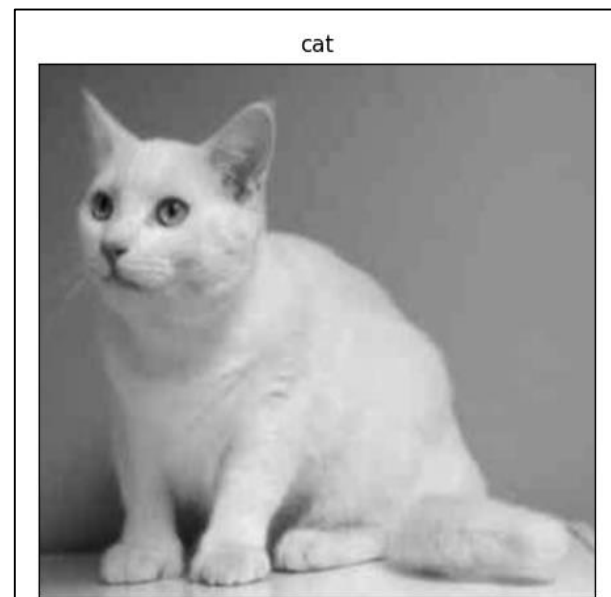
## ▪ 1.3 예제 3 - Matplotlib를 이용한 이미지 출력

```
import numpy as np
import cv2
import matplotlib.pyplot as plt

def showImage():
    imgfile = 'images/cat.jpg'
    img = cv2.imread(imgfile, cv2.IMREAD_GRAYSCALE)

    plt.imshow(img, cmap = 'gray', interpolation = 'bicubic')
    plt.xticks([])
    plt.yticks([])
    plt.title('cat')
    plt.show()

showImage()
```



[그림] 예제 3 실행 결과

# OpenCV 이미지 예제

- 1.3 예제 3 - Matplotlib를 이용한 이미지 출력

- Matplotlib은 파이썬을 위한 플로팅 라이브러리로서 다양한 플로팅 메소드를 제공한다. Matplotlib을 이용해 이미지를 화면에 나타낼 수 있으며 이미지 zoom, 저장하기 등과 같은 다양한 기능 등을 활용할 수 있다.

# OpenCV 이미지 예제

## ▪ 1.3 예제 3 - Matplotlib를 이용한 이미지 출력

```
>>img = cv2.imread(imgfile, cv2.IMREAD_GRAYSCALE)
```

- 이미지를 흑백으로 읽어온다. OpenCV는 BGR 모드로 컬러 이미지를 다루는데 비해 Matplotlib은 RGB 모드로 컬러 이미지를 다룬다. 따라서 OpenCV로 읽어 들인 컬러 이미지를 Matplotlib에 그대로 사용하게 될 경우 제대로 된 컬러 색상이 나오지 않는다.

```
>>plt.imshow(img, cmap='gray', interpolation='bicubic')
```

- Matplotlib의 imshow 함수를 이용하여 화면에 이미지를 디스플레이 하는 방법을 정의한다.

# OpenCV 이미지 예제

- 1.3 예제 3 - Matplotlib를 이용한 이미지 출력

```
>>plt.xticks([]), plt.yticks([])
```

- Matplotlib은 기본적으로 이미지를 표시할 때 x축 y축으로 눈금 표시를 한다. 위 코드는 눈금 표시 없이 이미지를 표시하라는 의미이다.

```
>>plt.show()
```

- 화면에 이미지를 출력한다.