

OpenCV 주요 함수

Mat imread(const String& filename, int flags = IMREAD_COLOR);

- filename : 불러올 영상 파일의 이름
 - 파일의 이름만 지정할 경우 프로그램 폴더 내에 위치한 파일을 불러옴.
 - 다른 폴더의 파일을 불러오려면 '절대 경로' 또는 '상대 경로' 형식으로 파일 위치 지정.
- flags : 영상 파일 불러오기 옵션 플래그. **ImreadModes 열거형 상수**를 지정
 - 영상 파일을 불러올 때 사용할 컬러 모드와 영상 크기를 지정
 - ImreadModes 열거형 상수를 지정 가능.

ImreadModes 열거형 상수	설명
IMREAD_UNCHANGED	입력 파일에 지정된 그대로의 컬러 속성 사용.
IMREAD_GRAYSCALE	1채널 그레이스케일 영상으로 변환하여 불러옴.
IMREAD_COLOR	3채널 BGR컬러 영상으로 변환하여 불러옴. (default 값)
IMREAD_REDUCED_GRAYSCALE_2	크기를 1/2로 줄인 1채널 그레이스케일 영상.
IMREAD_REDUCED_COLOR_2	크기를 1/2로 줄인 3채널 BGR 영상
IMREAD_IGNORE_ORIENTATION	EXIF에 저장된 방향 정보를 사용하지 않음.

- 반환값 : 불러온 영상 데이터(Mat 객체)
 - filename으로 지정된 파일이 존재하지 않거나 잘못된 형식의 영상파일이라면 비어있는 Mat 객체를 반환.
 - imread() 함수 사용 후 Mat::empty()를 이용하여 Mat 객체가 제대로 생성되었는지 확인할 것.

**bool imwrite(const String& filename, InputArray img,
const std::vector<int>& params = std::vector<int>());**

- filename : 저장할 영상 파일 이름
 - imwrite() 함수는 img 변수에 저장되어 있는 영상 데이터를 filename 이름의 파일로 저장함.
 - 영상 파일 형식은 filename 문자열에 포함된 파일 확장자를 따름.
- img : 저장할 영상 데이터 (Mat 객체)
- params : 저장할 영상 파일 형식에 의존적인 파라미터(플래그 & 값) 쌍
 - params 인자의 형식은 **std::vector<int>** 타입으로 지정
 - 옵션 플래그와 실제 값을 정수 값 두 개의 쌍으로 지정

- ex) img 변수에 저장된 영상을 lenna.jpg 파일로 저장할 때 JPEG 압축률을 95%로 지정하고 싶을 때의 params 인자 코드

```
vector<int> params;
params.push_back(IMWRITE_JPEG_QUALITY);
params.push_back(95);
imwrite("lenna.jpg", img, params);
```
- 위 코드에서 IMWRITE_JPEG_QUALITY 플래그가 JPEG 압축률을 의미하는 옵션 플래그.
- 반환값 : 정상적으로 저장하면 TRUE, 실패하면 FALSE 반환

bool Mat::empty() const

- 반환값 : 행렬의 rows 또는 cols 멤버 변수가 0이거나, data 멤버 변수가 NULL이면 true를 반환.

void namedWindow(const String& winname, int flags = WINDOW_AUTOSIZE);

- winname : 영상 출력 창 상단에 출력되는 창 고유 이름. 이 문자열로 창을 구분함.
- Windows 운영 체제는 각각의 창을 구분하기 위해 handle이라는 숫자 값을 사용.
- OpenCV에서는 각각의 창에 고유한 문자열을 부여하여 각각의 창을 구분.
- flags : 생성되는 창의 속성을 지정하는 플래그. **WindowFlags 열거형 상수**를 지정.

WindowFlags 열거형 상수	설명
WINDOW_NORMAL	영상 출력 창의 크기에 맞게 영상 크기가 변경되어 출력. 사용자 임의로 창 크기 변경 가능
WINDOW_AUTOSIZE	출력하는 영상 크기에 맞게 창 크기 자동 설정. 사용자 임의로 창 크기 변경 불가능 (default 값)
WINDOW_OPENGL	OpenGL 지원

- 현재 사용 중인 모니터 해상도보다 큰 영상을 화면에 출력하려고 할 경우, WINDOW_AUTOSIZE 속성으로 생성된 창에서는 영상의 일부가 화면에 표시되지 않을 수 있음.
- 생성한 창 크기를 마우스 또는 `resizeWindow()` 함수를 이용하여 변경을 원한다면 flags 인자에 WINDOW_NORMAL을 지정해야함.

void destroyWindow(const String& winname);
void destroyAllWindows();

- winname : 소멸 시킬 창 이름.

- 일반적으로 OpenCV 응용 프로그램이 완전히 종료될 때 운영 체제에 의해 OpenCV 응용 프로그램이 사용하던 모든 자원이 해제됨.
- 이 때, `namedWindow()` 함수에 의해 만들어진 창도 모두 자동으로 닫힘.
- 하지만 프로그램 동작 중에 창을 닫고 싶을 때에는 `destroyWindow()` 나 `destroyAllWindows()` 함수를 사용함.

`void moveWindow(const String& winname, int x, int y);`

- `winname` : 위치를 이동할 창의 이름
 - `winname`의 이름을 가진 창을 (x, y) 좌표 위치로 이동시킴.
 - 이때 좌표는 모니터 전체 화면에서의 좌표, 모니터 좌측 상단을 원점으로 함.
- `x` : 창이 이동할 위치의 x 좌표
- `y` : 창이 이동할 위치의 y 좌표

`void resizeWindow(const String& winname, int width, int height);`

- `winname` : 크기를 변경할 창 이름
 - `winname`을 가진 창의 크기를 가로 `width`, 세로 `height` 크기에 맞게 변경함.
 - 인자로 전달되는 `width`, `height` 값은 창 전체 크기가 아닌 창의 view 영역에 나타나는 영상 크기를 의미함.
 - 따라서 `resizeWindow()` 함수에 의해 변경된 창 크기는 제목 표시줄과 경계선 두께로 인해 전달한 인자의 값보다 조금 더 크게 됨.
 - `WINDOW_AUTOSIZE` 플래그를 사용하여 만들어진 영상 출력창은 이 함수로 크기 조절이 불가능함.
- `width` : 창의 가로 크기
- `height` : 창의 세로 크기

`void imshow(const String& winname, InputArray mat);`

- `winname` : 영상을 출력할 대상 창 이름
- `mat` : 출력할 영상 데이터(Mat 객체)
- `winname`의 이름을 가진 창에 `mat` 인자로 전달된 영상 데이터를 출력함.

mat 객체에 저장된 영상 데이터 타입	imshow()의 출력 형태
1채널 8비트 uchar 자료형(그레이스케일)	픽셀 값 그대로 그레이스케일 밝기 형태로 출력
3채널 uchar 컬러 영상	색상 채널이 Blue, Green, Red 순서로 되어 있다고 간주하여 색상 표현
부호 없는 16, 32 비트 정수형	행렬 원소 값을 256으로 나눈 값을 영상의 밝기 값으로 사용.
32, 64 비트 실수형 행렬	행렬 원소에 255를 곱한 값을 밝기 값으로 사용.

- InputArray 타입은 Mat, Vector<T> 등 다양한 객체를 표현할 수 있는 인터페이스 클래스임.

- 주로 OpenCV 함수 입력에 해당하는 인자의 자료형으로 사용됨. 즉, InputArray 클래스의 인자에 Mat 클래스 타입의 변수를 전달함.

<p style="text-align: center;">int waitKey(int delay = 0);</p>
<ul style="list-style-type: none"> • delay : 키 입력을 기다릴 시간(밀리초 단위). $\text{delay} \leq 0$ 이면 무한히 기다림. • 반환값 : 눌린 키의 ASCII code 값. 지정 시간 동안 키가 눌리지 않았으면 -1을 반환. <p>- imshow() 함수만으로는 화면에 영상이 나타나지 않음.</p> <p>- imshow() 함수 호출 이후 waitKey() 함수를 호출해야 화면 그리기 이벤트가 동작하여 화면에 정상적으로 출력됨.</p> <p>- 즉, imshow() 함수와 waitKey() 함수는 연속하여 호출하는 형태로 사용.</p>