

KANGWON NATIONAL UNIVERSITY

# 컴퓨터비전 실습

실습2 | Mat

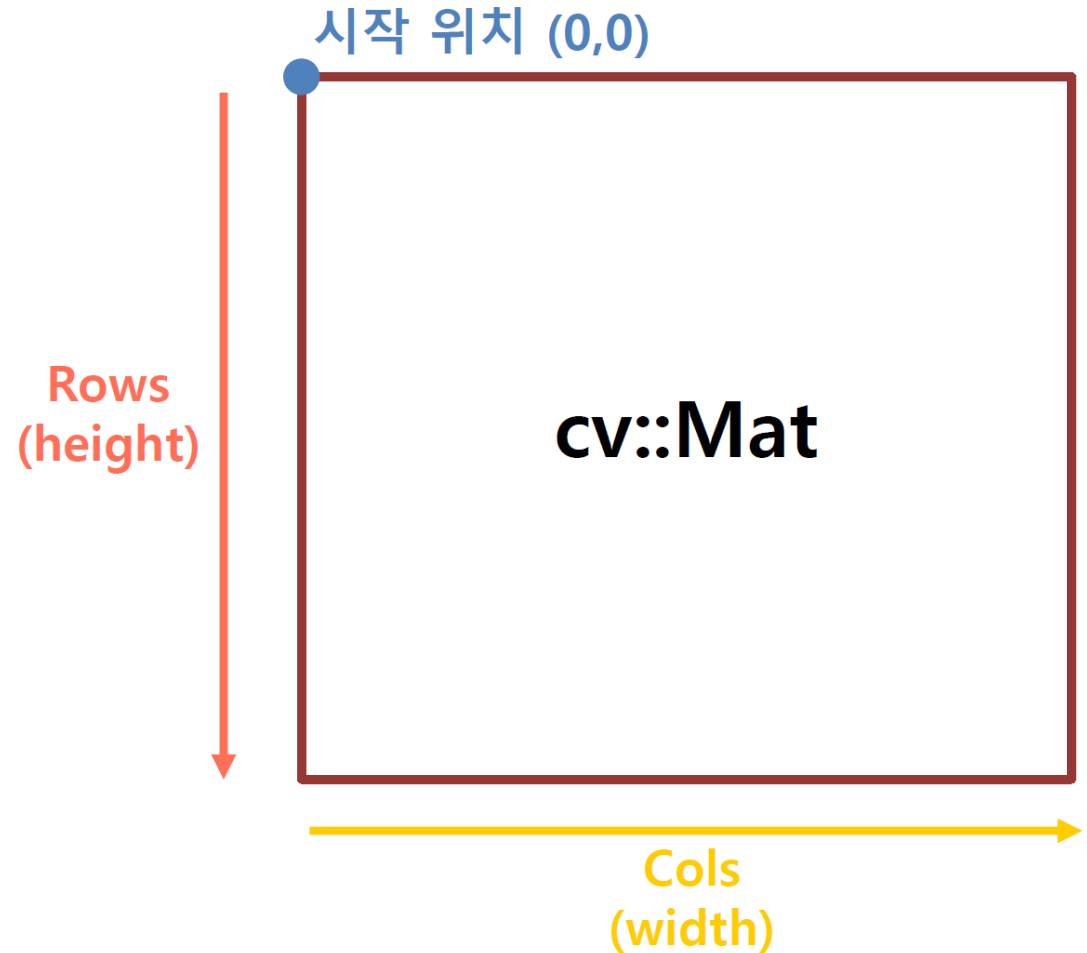
실습기간 3/19 ~ 3/25 (PM 23:59)

실습과제 이메일 제출

CVMIPLAB @ KNU

# Mat

- OpenCV에서 사용하는 이미지 단위
  - 왼쪽 상단부터 시작하여,  
왼쪽에서 오른쪽 방향으로,  
위에서 아래 방향으로 픽셀 값을 저장함
  - Mat 객체에 접근하여  
이미지를 픽셀 단위나 채널 단위로  
값을 확인하거나 바꿀 수 있음



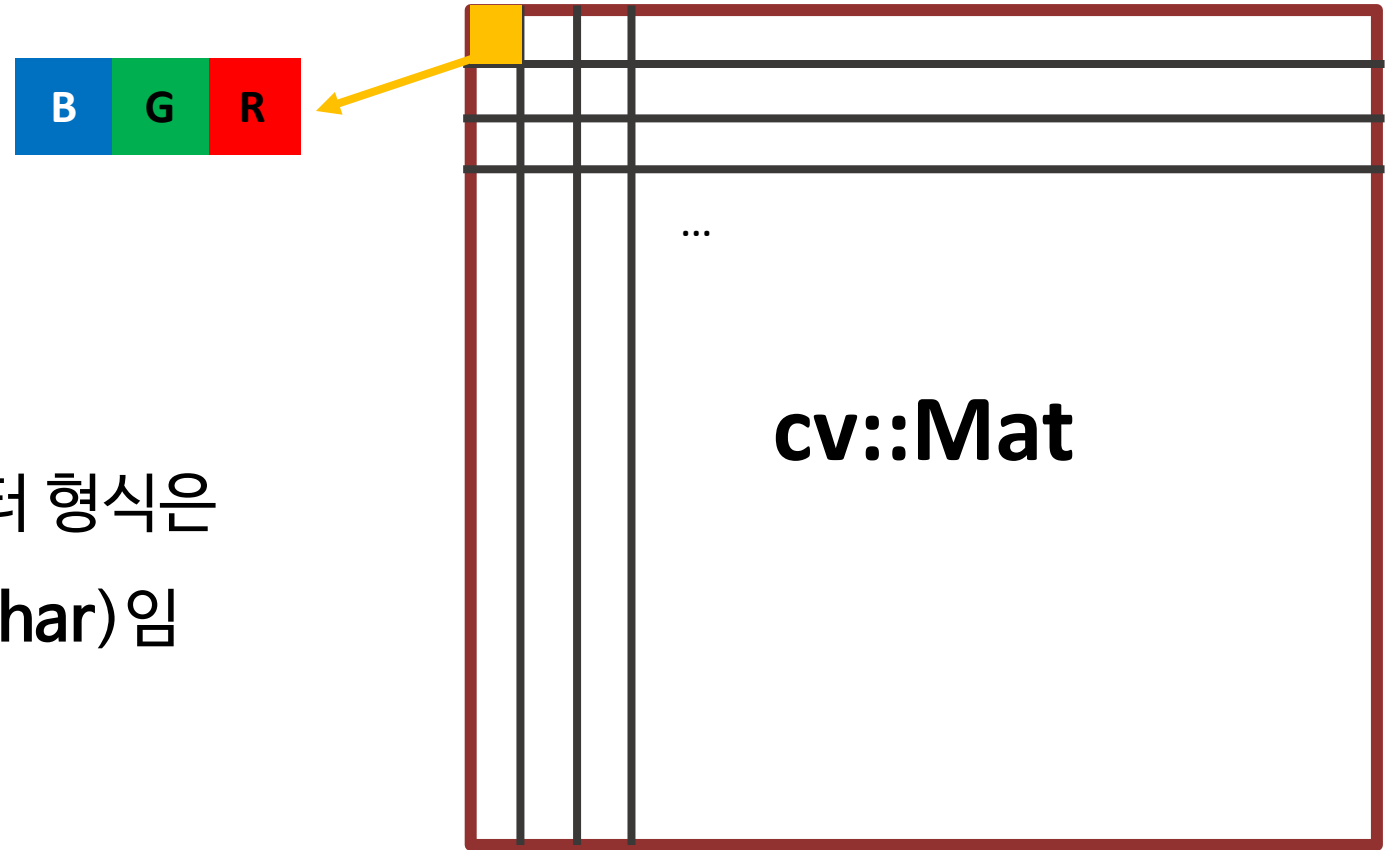
# Mat

- 이미지 저장 단위

- 각 픽셀마다 B,G,R 의 순서로 값이 저장되어 있음

**(R,G,B가 아님!!)**

- 기본적으로 값이 저장되는 데이터 형식은 8비트인 `unsigned char` 형 (`uchar`) 임 (범위: 0~255)



# Mat

- 이미지 저장 단위

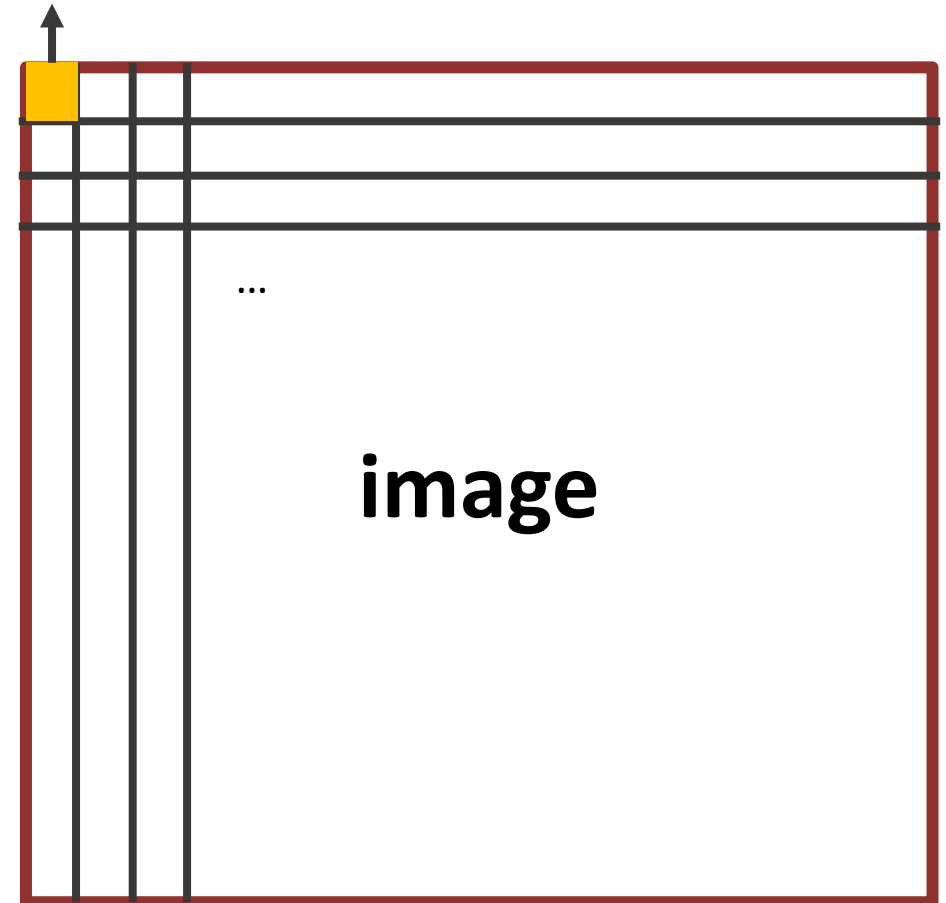
```
1  #include "opencv2/opencv.hpp"
2
3  using namespace cv;
4
5  int main()
6  {
7      // 이미지를 저장할 변수를 선언한다.
8      cv::Mat image;
9
10     // cat.bmp 파일을 읽어 할당한다.
11     image = cv::imread("cat.bmp", cv::IMREAD_COLOR);
12
13     // 이미지 최상단 좌측의 픽셀 한 개의 R, G, B값을 읽고 출력한다.
14     int x = 0, y = 0;
15     std::cout << "B: " << (unsigned int)image.at<Vec3b>(y, x)[0] << " ";
16     std::cout << "G: " << (unsigned int)image.at<Vec3b>(y, x)[1] << " ";
17     std::cout << "R: " << (unsigned int)image.at<Vec3b>(y, x)[2] << std::endl;
18
19     return 0;
20 }
```



B: 56 G: 49 R: 44

※ int형으로 casting하지 않고 출력시 char형으로 출력된다

`image.at<Vec3b>(0, 0) = {56, 49, 44}`



- 컬러 표현 (BGR)

- 배열의 각 인덱스로  각각의 요소에 접근할 수 있다.
  - `image.at<Vec3b>(y, x)[0]` // 첫번째 컬러 도메인 (B)
  - `image.at<Vec3b>(y, x)[1]` // 두번째 컬러 도메인 (G)
  - `image.at<Vec3b>(y, x)[2]` // 세번째 컬러 도메인 (R)

- 그레이스케일 표현 (Grayscale)

- 영상을 파일에서 로드할 때, 그레이스케일로 로드할 수 있다.

```
image = cv::imread("cat.bmp", cv::IMREAD_GRAYSCALE);
```

- 그레이스케일 영상을 읽을 경우 픽셀마다 컬러 도메인이 1개이다.

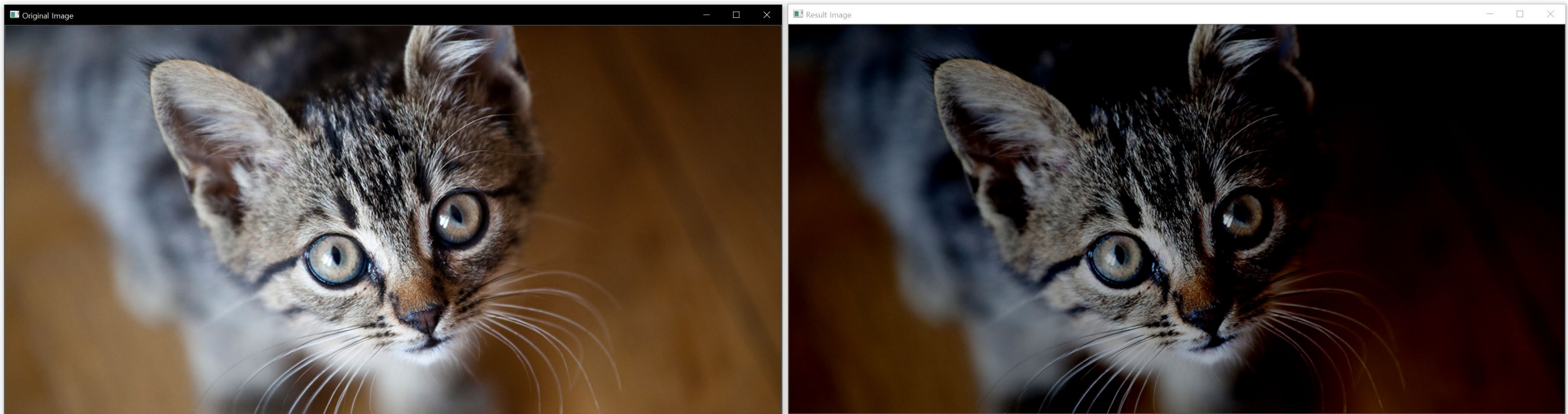
→ `image.at<uchar>(y, x)`    // 첫번째 컬러 도메인 (Intensity)  
    // uchar 타입으로 접근하며 배열없이 바로 원소에 접근한다.

- 이미지 연산 예제 - 밝기 조절
  - 이미지를 읽어와 밝기를 조절하여 출력한다.
  - `cv::saturate_cast<uchar>(a)`
    - 변수 **a**를 uchar 값의 범위 (0~255 사이)로 clip한다.  
(예: a=280일때 결과 → a=255,  
a=-16일때 결과 → a=0)

```
1  #include "opencv2/opencv.hpp"
2
3  using namespace cv;
4
5  int main()
6  {
7      // 이미지를 저장할 변수를 선언한다.
8      cv::Mat image, result_image;
9
10     // cat.bmp 파일을 읽어 할당한다.
11     // result_image에도 같은 내용을 할당한다.
12     // → result_image에 이미지 크기 정보를 할당하기 위한 코드이다.
13     image = cv::imread("cat.bmp", cv::IMREAD_COLOR);
14     image.copyTo(result_image);
15
16     // 이미지의 전체 밝기 값을 10 낮춘다.
17     for (int y = 0; y < image.rows; y++)
18     {
19         for (int x = 0; x < image.cols; x++)
20         {
21             // image로부터 픽셀 값을 읽어옴과 동시에 수정한다.
22             int b = image.at<Vec3b>(y, x)[0] - 75;
23             int g = image.at<Vec3b>(y, x)[1] - 75;
24             int r = image.at<Vec3b>(y, x)[2] - 75;
25
26             // 값을 0~255 범위에 맞게 clip하여 result_image에 할당한다.
27             result_image.at<Vec3b>(y, x)[0] = cv::saturate_cast<uchar>(b);
28             result_image.at<Vec3b>(y, x)[1] = cv::saturate_cast<uchar>(g);
29             result_image.at<Vec3b>(y, x)[2] = cv::saturate_cast<uchar>(r);
30         }
31     }
32
33     // 이미지를 서로 비교한다.
34     cv::imshow("Original Image", image);
35     cv::imshow("Result Image", result_image);
36     cv::waitKey(0); // 아무 키나 누르면 종료된다.
37
38     return 0;
39 }
```

# Mat

- 이미지 연산 예제 - 밝기 조절 결과 화면





# 과제 제출

- 연락처/문의방법

- 컴퓨터비전 TA: 안정인 / 컴퓨터비전&의료영상처리연구실
- 이루리 내 Q&A 게시판에 질문 작성 또는  
이메일 [ji5489@gmail.com](mailto:ji5489@gmail.com)으로 메일제목 “[컴퓨터비전] 질문”으로 문의
- 공학6호관 509호 (방문 시 마스크를 꼭 착용해주시기 바랍니다.)

- 과제 제출 방법

- ~~이메일 [ji5489@gmail.com](mailto:ji5489@gmail.com)으로 메일제목 “[컴퓨터비전] n주차과제”로 제출  
(n=PPT 표지에 있는 주차번호, 띄어쓰기 없음)~~ **이루리 내 과제 제출란에 압축하여 제출합니다.**

# 실습 2-1 | 이미지 반전

## 문제

주어진 “.bmp” 파일을 읽고, 이미지를 반전합니다.

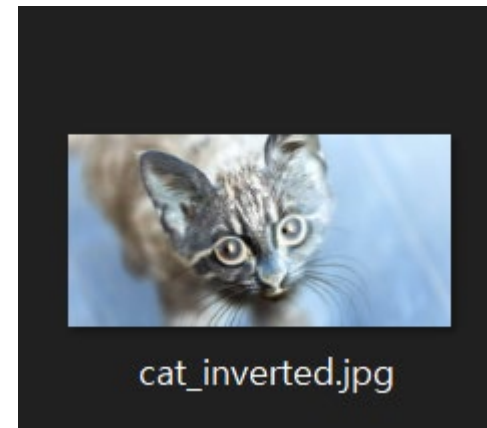
반전된 이미지를 파일로 저장하세요. 저장 파일명은 “cat\_inverted.jpg”입니다.

(“이미지 연산 예제”와 강의 “3강 영상처리 기초 - 화질개선 ①”을 참고하여 과제를 수행합니다.)

## 요구 결과

결과 파일 “cat\_inverted.jpg”과 main함수가 포함된 “.cpp” 코드 두 개를 압축해 제출합니다.

## 실행결과



# 실습 2-2 | 이진화 구현

## 문제

주어진 “.bmp” 파일을 **흑백으로** 읽고, 이미지를 이진화하여 파일 “cat\_binarized.jpg”로 저장합니다.

여기서 이진화란 0부터 255까지로 분포된 값들을 0과 255 두 가지 값만 가지도록 하는 연산으로 정의합니다.

(보통은 0과 1로 변환하는 작업을 의미하나 편의상 0과 255로 하도록 하겠습니다.)

다음 수식에 따라 이진화를 수행하세요:  $P^{y,x} = \{P_B^{y,x}, P_G^{y,x}, P_R^{y,x}\}$  일 때,  $bin(P^{y,x}) = \begin{cases} 0 & (P^{y,x} < 128) \\ 255 & (P^{y,x} \geq 128) \end{cases}$

## 요구 결과

결과 파일 “cat\_binarized.jpg”과 main함수가 포함된 “.cpp” 코드 두 개를 압축해 제출합니다.

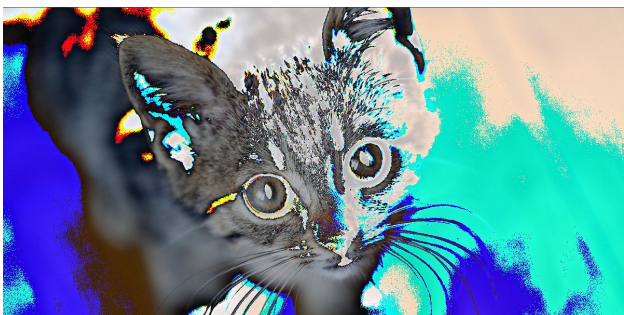
## 실행결과 및 코드

참고: 파일을 흑백으로 읽기 위해서는 IMREAD\_GRAYSCALE을 사용합니다.



# Q&A

## Q1. 결과 이미지가 깨져서 나오는 경우



→ 각 픽셀의 B, G, R 값은 0~255 사이의 범위를 가집니다.  
범위가 넘어가면 underflow나 overflow 문제가 발생합니다.  
해당 문제를 해결하기 위해 더 큰 자료형에 담아두고 계산한 뒤,  
`cv::saturate_cast<uchar>` 함수를 사용합니다.

## Q2. `Error: Assertion failed (size.width>0 && size.height>0)`

→ 이미지 파일이 디렉토리에 있고 정상적으로 불러왔는지 확인해봅니다.