KANGWON NATIONAL UNIVERSITY

컴퓨터비전 실습

실습3 | Filtering

실습기간 3/26 ~4/1 (PM 23:59)
실습과제 이루리 내 제출

CVMIPLAB @ KNU

모범답안 2-1 | 이미지 반전

문제

주어진 “.bmp” 파일을 읽고, 이미지를 반전합니다.

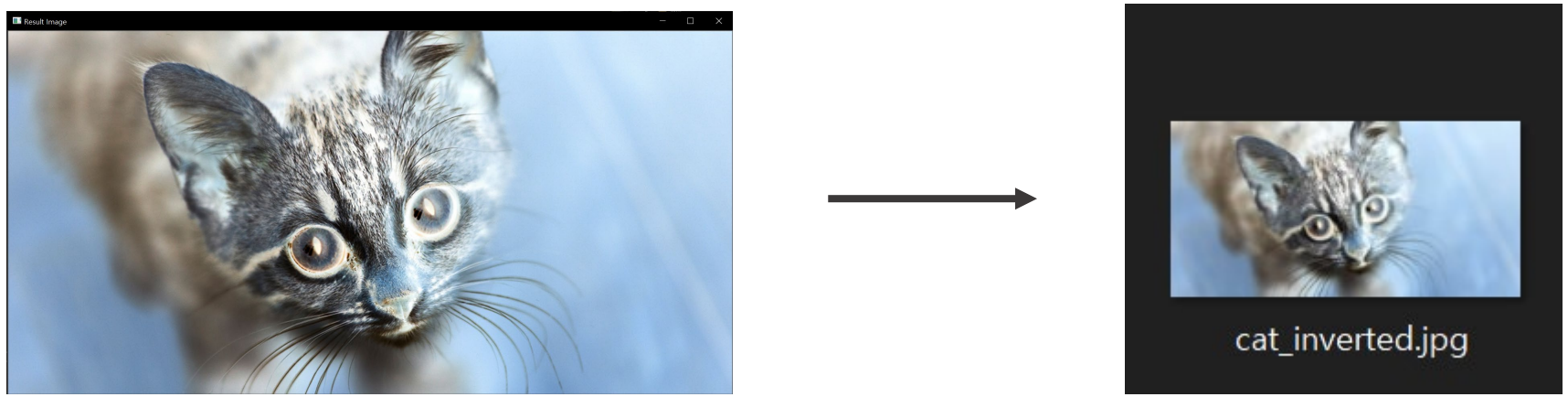
반전된 이미지를 파일로 저장하세요. 저장 파일명은 “cat_inverted.jpg”입니다.

(“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



모범답안 2-1 | 이미지 반전

```cpp
/*
    과제2-1: 이미지 반전

    @author 안정인
    @license 2021 CVMIPLab
    @date 2021-03-18
*/

#include "opencv2/opencv.hpp"

int main()
{
    // 이미지를 저장할 변수를 선언한다.
    cv::Mat image, result_image;

    // cat.bmp 파일을 읽어 할당한다.
    // result_image에도 같은 내용을 할당한다.
    image = cv::imread("cat.bmp", cv::IMREAD_COLOR);
    image.copyTo(result_image);  // → result_image에 이미지 크기 정보를 할당하기 위한 코드이다.

    // 이미지의 전체 밝기 값을 10 낮춘다.
    for (int y = 0; y < image.rows; y++)
    {
        for (int x = 0; x < image.cols; x++)
        {
            // image로부터 픽셀 값을 읽어옴과 동시에 반전한다.
            int b = 255 - image.at<cv::Vec3b>(y, x)[0];
            int g = 255 - image.at<cv::Vec3b>(y, x)[1];
            int r = 255 - image.at<cv::Vec3b>(y, x)[2];

            // 값을 0~255 범위에 맞게 clip하여 result_image에 할당한다.
            result_image.at<cv::Vec3b>(y, x)[0] = cv::saturate_cast<uchar>(b);
            result_image.at<cv::Vec3b>(y, x)[1] = cv::saturate_cast<uchar>(g);
            result_image.at<cv::Vec3b>(y, x)[2] = cv::saturate_cast<uchar>(r);
        }
    }
```

```cpp

    cv::imshow("Result Image", result_image);
    cv::waitKey(0); // 아무 키나 누르면 계속한다.

    // 반전된 이미지를 파일로 저장한다.
    cv::imwrite("cat_inverted.jpg", result_image);
    std::cout << "파일 저장 완료!" << std::endl;

    return 0;
}
```

모범답안 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.

모범답안 2-2 | 이진화 구현

```cpp
/*
    과제2-2: 이미지 이진화

    @author 안정인
    @license 2021 CVMIPLab
    @date 2021-03-19
*/

#include "opencv2/opencv.hpp"

int main()
{
    // 이미지를 저장할 변수를 선언한다.
    cv::Mat image, result_image;

    // cat.bmp 파일을 읽어 할당한다.
    // result_image에도 같은 내용을 할당한다.
    image = cv::imread("cat.bmp", cv::IMREAD_GRAYSCALE);
    image.copyTo(result_image);  // → result_image에 이미지 크기 정보를 할당하기 위한 코드이다.
```

```cpp
    int i;
    for (int y = 0; y < image.rows; y++)
    {
        for (int x = 0; x < image.cols; x++)
        {
            // image로부터 픽셀 값을 읽어오고 이를 이진화한다.
            // 값을 0~255 범위에 맞게 clip하여 result_image에 할당한다.
            i = image.at<uchar>(y, x);
            i = (i < 128) ? 0 : 255;
            result_image.at<uchar>(y, x) = cv::saturate_cast<uchar>(i);
        }
    }

    cv::imshow("Result Image", result_image);
    cv::waitKey(0); // 아무 키나 누르면 계속한다.

    // 이진화된 이미지를 파일로 저장한다.
    cv::imwrite("cat_inverted.jpg", result_image);
    std::cout << "파일 저장 완료!" << std::endl;

    return 0;
}
```

과제 제출

- 연락처/문의방법
  - 컴퓨터비전 TA: 안정인 / 컴퓨터비전&의료영상처리연구실
  - 이루리 내 Q&A 게시판에 질문 작성 또는

    이메일 ji5489@gmail.com으로 메일제목 “[컴퓨터비전]질문”으로 문의
  - 공학6호관 509호 (방문 시 마스크를 꼭 착용해주시기 바랍니다.)
- 과제 제출 방법
  - 이루리 내 과제 제출란에 압축하여 제출합니다.

실습 3-1 | Dissolve 구현

문제

주어진 두 “cat.bmp”, “tibetfox.bmp” 파일을 읽고, 아래 수식에 따라 디졸브를 구현합니다.

$$f_{out}(j,i) = \alpha f_1(j,i) + (1-\alpha) f_2(j,i)$$

디졸브 시, $\alpha = 0.3$일 때와 $\alpha = 0.7$일 때를 각각 “dissolve_3.bmp”, “dissolve_7.bmp”로 제출합니다.

※ 다음 슬라이드의 실행 결과를 보고 디졸브 순서에 유의하세요.

다만 영상과 같은 **실시간으로 움직이는 시퀀스를 구현하지 않아도 됩니다 (파일 두 개만 생성하도록 해도 됩니다.)**

요구 결과

결과 파일 “dissolve_3.bmp”, “dissolve_7.bmp”과

main함수가 포함된 “.cpp” 코드 총 3개를 압축해 제출합니다.

실습 3-1 | Dissolve 구현

실행결과 동영상

실습 3-2 | Low Pass Filter 구현

문제

주어진 “salt_pepper.bmp” 파일을 읽고, Low Pass Filter를 구현하여 적용한 뒤 “salt_pepper_lpf.bmp”로 저장하세요.

Low pass filter 계수:

$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$
$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$
$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$

요구 결과

결과 파일 “salt_pepper_lpf.bmp”와 main함수가 포함된 “.cpp” 코드 두 개를 압축해 제출합니다.

실행결과 및 코드

※ 필터를 적용할 때는 Element-wise로 곱하여 구현하고, **`cv::filter2D`를 사용하면 안 됩니다.**

※ 필터를 정의할 때는 일반 배열로 정의하는 것을 권장합니다.

실습 3-2 | Low Pass Filter 구현

실행결과

실습 3-3 | High Pass Filter 구현

문제

주어진 “tibetfox.bmp” 파일을 읽고, High Pass Filter를 구현하여 적용한 뒤 “tibetfox _hpf.bmp”로 저장하세요.

High pass filter 계수:

−1	−1	−1
−1	8	−1
−1	−1	−1

요구 결과

결과 파일 “tibetfox _hpf.bmp”와 main함수가 포함된 “.cpp” 코드 두 개를 압축해 제출합니다.

실행결과 및 코드

※ 필터를 적용할 때는 Element-wise로 곱하여 구현하고, **`cv::filter2D`를 사용하면 안 됩니다.**

※ 필터를 정의할 때는 일반 배열로 정의하는 것을 권장합니다.

실습 3-3 | High Pass Filter 구현

실행결과