**디지털영상처리 실습 보고서**

이 름 : 박도령

학 번 : 2010112406

주 차 : 2주차

작성 일자 : 2016. 3. 14

제출 일자 : 2016. 3. 17

**문제 1)**

OpenCV 라이브러리를 이용하여 코드를 작성하여 윈도우 창으로 이미지를 띄우고 나머지 윈도우 창에 그 이미지를 **좌우 반전**한 새로운 이미지를 같이 띄워라.

♋ **사용 IDE 및 OpenCV 설치 버전**

**Visual Studio Community 2015 / OpenCV 3.1.0 버전**

♋ **소스 코드(ReportCode1.cpp)**

#include<opencv2/core/core.hpp>

#include<opencv2/highgui/highgui.hpp>

#include<opencv2/imgproc/imgproc.hpp>

#include<iostream>

int main(int argc, char\* argv[])

{

char \*imgFileName = "image.jpg";

char \*winNameIn = "original image";

char \*winNameOut = "processed image";

IplImage \*pImgIpl = 0, \*pImgIplOut;

// Create an image

pImgIpl = cvLoadImage(imgFileName);

// Copy the image data

pImgIplOut = cvCloneImage(pImgIpl);

if (!pImgIpl)

printf("can't load image: %s\n", imgFileName);

else

{

// Create a window

cvNamedWindow(winNameIn, CV\_WINDOW\_AUTOSIZE);

cvNamedWindow(winNameOut, CV\_WINDOW\_AUTOSIZE);

// Move Window

cvMoveWindow(winNameIn, 50, 50);

cvMoveWindow(winNameOut, 235, 50);

// Flip the image

**cvFlip(pImgIpl, pImgIplOut, 1);**

// Show the image

cvShowImage(winNameIn, pImgIpl);

cvShowImage(winNameOut, pImgIplOut);

// Waiting keyboard input

cvWaitKey(0);

// Store the image

cvSaveImage("processed.jpg", pImgIplOut);

// Release the image

cvReleaseImage(&pImgIpl);

cvReleaseImage(&pImgIplOut);

// Destroy the window

cvDestroyWindow(winNameIn);

cvDestroyWindow(winNameOut);

}

}

♋ **동작 순서 및 원리**

0. IplImage 객체를 이용하여 이미지를 하나의 객체로 처리할 수 있다. cvLoadImage 함수에 매개 변수로 이미지 파일을 주면 해당 이미지 객체에 대한 포인터로 사용할 수 있다. 매개 변수로는 다음과 같다.

**cvLoadImage(이미지 파일 경로)**

1. cvCloneImage 함수에 이미지 객체를 매개 변수로 주면 똑같은 이미지 객체에 대한 포인터를 얻을 수 있다.

**cvCloneImage(이미지 객체 포인터)**

2. 이미지를 처리하는 데에 있어서 기본적인 코드 형태는 다음과 같다.

**If(!이미지 객체 포인터){**

**오류 출력**

**}else{**

**해당 이미지 객체 처리 코드**

**}**

만약 이미지가 정상적으로 로드 되지 않았을 경우 해당 이미지 객체 포인터가 0을 반환한다. 따라서 이런 경우사용자가 오류에 대한 처리를 할 수 있게 if문 조건에 걸러질 수 있다. 정상적으로 로드 되었을 경우는 나머지 경우이므로 이 괄호 안에서는 로드된 이미지를 처리하는 코드를 넣을 수 있다.

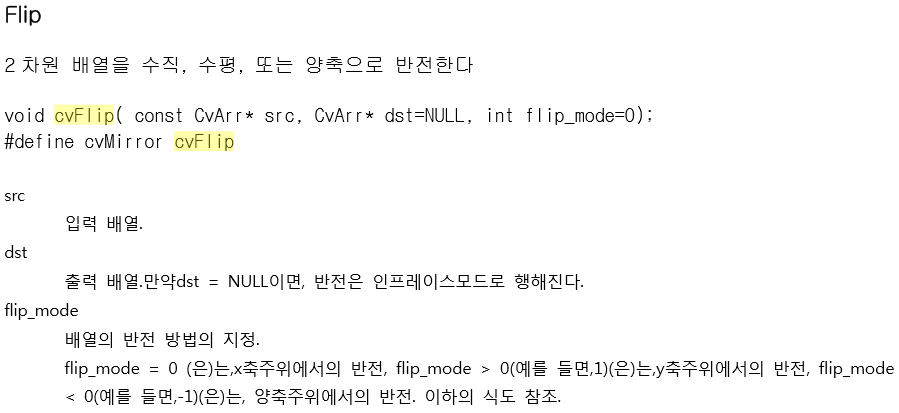
3. 이미지를 띄우는 수단으로는 윈도우가 필요하다. 따라서 윈도우 객체를 생성한다. 원본 이미지와 좌우 반전된 이미지를 띄울 것이므로 윈도우 객체 2개를 생성한다. 매개 변수로는 다음과 같다.

**cvNamedWindow(윈도우 이름, 윈도우 사이즈)**

4. 생성한 윈도우들의 생성 위치를 조정하기 위해서 cvMoveWindow 함수를 활용하였다. 매개변수로는 다음과 같다.

**cvMoveWindow(윈도우 이름, x좌표, y좌표)**

5. 함수 cvFlip을 이용하여 이미지를 좌우로 뒤집어 보았다. cvFilp 함수에 대한 매뉴얼은 다음과 같다.



좌우 반전은 y축 주위의 반전이므로 마지막 매개변수로 1을 주었다.

6. cvWaitKey 함수는 프로그램을 입력 대기 상태로 만든다. 따라서 이미지를 띄운 뒤 뒤의 이미지 해제 및 윈도우 종료 코드가 아직 실행되지 않게끔 하여 띄운 이미지를 계속 볼 수 있게끔 한다.

7. cvSaveImage 함수는 이미지를 파일의 형태로 저장하도록 하는 함수이다. 따라서 이미지가 해당 프로젝트 폴더에 저장된다. 매개 변수는 다음과 같다.

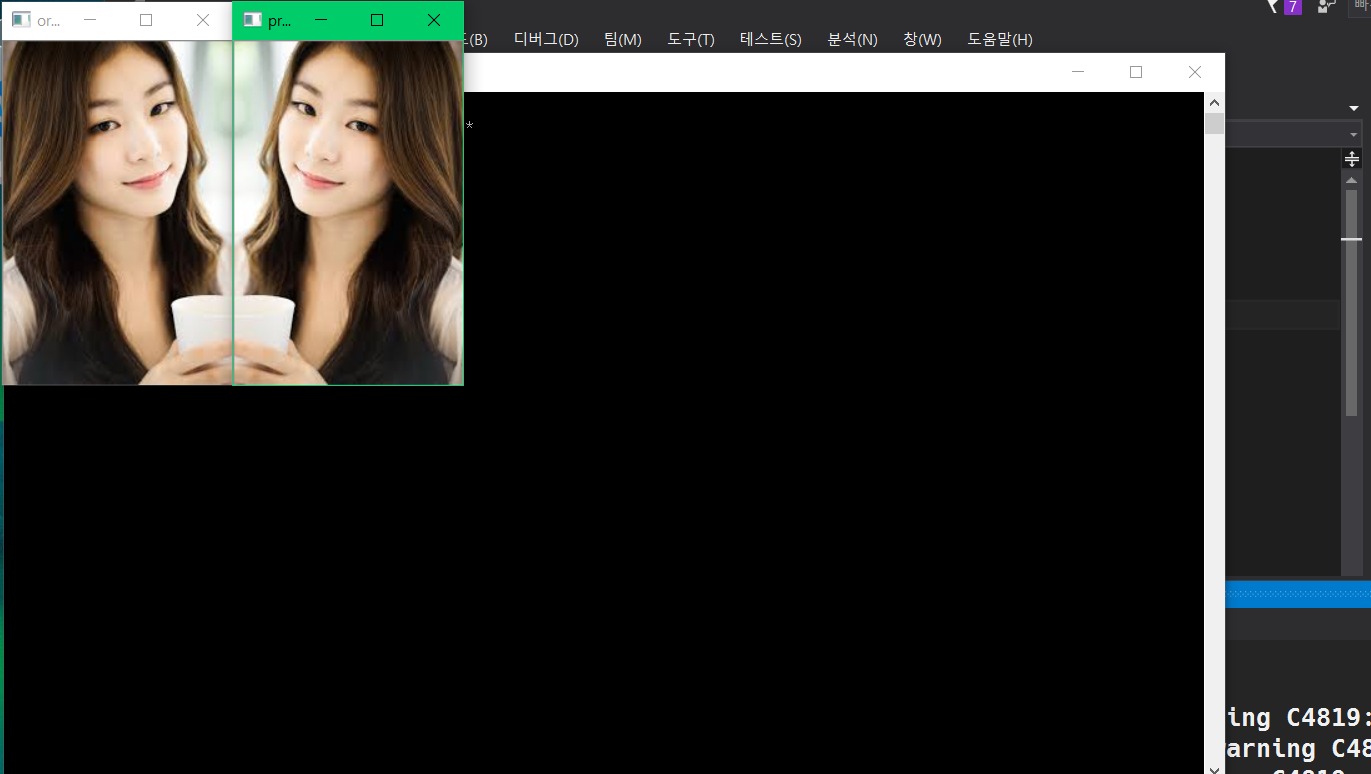
**cvSaveImage(저장할 이미지 이름, 저장할 이미지 객체 포인터)**

8. cvReleaseImage 함수와 cvDestroyWindow 함수는 프로그램 종료 직전에 반드시 사용해야 할 함수이다. cvRelaseImage는 로드된 이미지를 해제하는 함수이고, cvDestoryWindow는 생성된 윈도우를 해제하는 함수이다. 해당 객체들을 해제하지 않고 프로그램을 종료하면 문제가 발생할 수 있기 때문에 위 코드에서도 생성한 모든 이미지 객체와 윈도우를 이 함수를 이용하여 해제하였다.

**cvReleaseImage(이미지 객체의 이중 포인터), cvDestoryWindow(윈도우 이름)**

♋ **실행 화면**

**이미지는 좌우 반전을 명확하게 보여주기 위해 인물사진을 활용하였다.**



**저장된 이미지**

