Digital Image Processing

LabWork #CV1

Professor : Jin-Woo Jung

* Using the given CPixel class, design a program that can process the following :

1. Addition with two images
   1. Input :, stuff\_color\_2.png

|  |  |  |
| --- | --- | --- |
| **Input** | | **Output** |
|  |  |  |

| code |
| --- |
| #include <opencv2/opencv.hpp>  #include "Pixel.h"  using namespace cv;  int main()  {  CPixel processing; //pixel processing을 위한 클래스를 생성한다.  Mat image1 = imread("stuff\_color\_1.png", IMREAD\_COLOR); // stuff\_color\_1.png 를 불러온다. 이때 IMREAD\_COLOR 파라미터는 컬러 이미지를 불러온다는 것을 알려주기 위한 플래그 역할을 한다.  Mat image2 = imread("stuff\_color\_2.png", IMREAD\_COLOR); // stuff\_color\_2.png 를 불러온다. 이때 IMREAD\_COLOR 파라미터는 컬러 이미지를 불러온다는 것을 알려주기 위한 플래그 역할을 한다.  //추가로 imread 함수 두 번째 파라미터에는 IMREAD\_COLOR 외에 gray-scale로 읽어들이는 IMREAD\_GRAYSCALE과 alpha channel까지 포함하여 읽어들이는 IMREAD\_UNCHANGED가 올 수 있다.  Mat add = processing.GS\_add\_image(image1, image2); // 두 이미지를 이용해 addition 연산을 행한다.  imshow("image", add); //addition 연산한 이미지를 출력한다. 이때 첫번째 파라미터는 이미지를 출력하기 위해 띄우는 창의 제목표시줄에 출력할 제목이고, 두 번째 파라미터는 출력하기 위한 이미지를 가리키는 변수이다.  waitKey(); //키 입력이 있을 때까지 창을 유지하기 위해 waitkey() 함수를 이용한다.  return 0;  } |

* 1. Input : lenna.jpg, orange.jpg

|  |  |  |
| --- | --- | --- |
| Input | | Output |
|  |  |  |

| code |
| --- |
| #include <opencv2/opencv.hpp>  #include "Pixel.h"  using namespace cv;  int main()  {  CPixel processing; //pixel processing을 위한 클래스를 생성한다.  Mat image1 = imread("lenna.png", IMREAD\_COLOR); // lenna.png 를 불러온다. 이때 IMREAD\_COLOR 파라미터는 컬러 이미지를 불러온다는 것을 알려주기 위한 플래그 역할을 한다.  Mat image2 = imread("orange.jpg", IMREAD\_COLOR); // orange.jpg 를 불러온다. 이때 IMREAD\_COLOR 파라미터는 컬러 이미지를 불러온다는 것을 알려주기 위한 플래그 역할을 한다.  //추가로 imread 함수 두 번째 파라미터에는 IMREAD\_COLOR 외에 gray-scale로 읽어들이는 IMREAD\_GRAYSCALE과 alpha channel까지 포함하여 읽어들이는 IMREAD\_UNCHANGED가 올 수 있다.  Mat add = processing.GS\_add\_image(image1, image2); // 두 이미지를 이용해 addition 연산을 행한다.  imshow("image", add); //addition 연산한 이미지를 출력한다. 이때 첫번째 파라미터는 이미지를 출력하기 위해 띄우는 창의 제목표시줄에 출력할 제목이고, 두 번째 파라미터는 출력하기 위한 이미지를 가리키는 변수이다.  waitKey(); //키 입력이 있을 때까지 창을 유지하기 위해 waitkey() 함수를 이용한다.  return 0;  } |

1. Subtraction with two images
   1. Input : stuff\_color\_1.png, stuff\_color\_2.png

|  |  |  |
| --- | --- | --- |
| Input | | Output |
|  |  |  |
| code | | |
| #include <opencv2/opencv.hpp>  #include "Pixel.h"  using namespace cv;  int main()  {  CPixel processing; //pixel processing을 위한 클래스를 생성한다.  Mat image1 = imread("stuff\_color\_1.png", IMREAD\_COLOR); // stuff\_color\_1.png 를 불러온다. 이때 IMREAD\_COLOR 파라미터는 컬러 이미지를 불러온다는 것을 알려주기 위한 플래그 역할을 한다.  Mat image2 = imread("stuff\_color\_2.png", IMREAD\_COLOR); // stuff\_color\_2.png 를 불러온다. 이때 IMREAD\_COLOR 파라미터는 컬러 이미지를 불러온다는 것을 알려주기 위한 플래그 역할을 한다.  //추가로 imread 함수 두 번째 파라미터에는 IMREAD\_COLOR 외에 gray-scale로 읽어들이는 IMREAD\_GRAYSCALE과 alpha channel까지 포함하여 읽어들이는 IMREAD\_UNCHANGED가 올 수 있다.  Mat sub = processing.GS\_subtract\_image(image1, image2); // 두 이미지를 이용해 subtraction 연산을 행한다.  imshow("image", sub); // subtraction 연산한 이미지를 출력한다. 이때 첫번째 파라미터는 이미지를 출력하기 위해 띄우는 창의 제목표시줄에 출력할 제목이고, 두 번째 파라미터는 출력하기 위한 이미지를 가리키는 변수이다.  waitKey(); //키 입력이 있을 때까지 창을 유지하기 위해 waitkey() 함수를 이용한다.  return 0;  } | | |

* 1. Input : lenna.jpg, orange.jpg

|  |  |  |
| --- | --- | --- |
| Input | | Output |
|  |  |  |
| code | | |
| #include <opencv2/opencv.hpp>  #include "Pixel.h"  using namespace cv;  int main()  {  CPixel processing; //pixel processing을 위한 클래스를 생성한다.  Mat image1 = imread("lenna.png", IMREAD\_COLOR); // lenna.png 를 불러온다. 이때 IMREAD\_COLOR 파라미터는 컬러 이미지를 불러온다는 것을 알려주기 위한 플래그 역할을 한다.  Mat image2 = imread("orange.jpg", IMREAD\_COLOR); // orange.jpg 를 불러온다. 이때 IMREAD\_COLOR 파라미터는 컬러 이미지를 불러온다는 것을 알려주기 위한 플래그 역할을 한다.  //추가로 imread 함수 두 번째 파라미터에는 IMREAD\_COLOR 외에 gray-scale로 읽어들이는 IMREAD\_GRAYSCALE과 alpha channel까지 포함하여 읽어들이는 IMREAD\_UNCHANGED가 올 수 있다.  Mat sub = processing.GS\_subtract\_image(image1, image2); // 두 이미지를 이용해 subtraction 연산을 행한다.  imshow("image", sub); // subtraction 연산한 이미지를 출력한다. 이때 첫번째 파라미터는 이미지를 출력하기 위해 띄우는 창의 제목표시줄에 출력할 제목이고, 두 번째 파라미터는 출력하기 위한 이미지를 가리키는 변수이다.  waitKey(); //키 입력이 있을 때까지 창을 유지하기 위해 waitkey() 함수를 이용한다.  return 0;  } | | |

1. Blending with two images
   * 1. Input : stuff\_color\_1.png, stuff\_color\_2.png

|  |  |  |
| --- | --- | --- |
| **Input** | | **Output** |
|  |  |  |
| code | | |
| #include <opencv2/opencv.hpp>  #include "Pixel.h"  using namespace cv;  int main()  {  CPixel processing; //pixel processing을 위한 클래스를 생성한다.  Mat image1 = imread("stuff\_color\_1.png", IMREAD\_COLOR); // stuff\_color\_1.png 를 불러온다. 이때 IMREAD\_COLOR 파라미터는 컬러 이미지를 불러온다는 것을 알려주기 위한 플래그 역할을 한다.  Mat image2 = imread("stuff\_color\_2.png", IMREAD\_COLOR); // stuff\_color\_2.png 를 불러온다. 이때 IMREAD\_COLOR 파라미터는 컬러 이미지를 불러온다는 것을 알려주기 위한 플래그 역할을 한다.  //추가로 imread 함수 두 번째 파라미터에는 IMREAD\_COLOR 외에 gray-scale로 읽어들이는 IMREAD\_GRAYSCALE과 alpha channel까지 포함하여 읽어들이는 IMREAD\_UNCHANGED가 올 수 있다.  Mat blending = processing.GS\_blending\_effect(image1, image2, 0.5); // 두 이미지를 이용해 blending 연산을 행한다.이때 3번째 파라미터에는 0과 1 사이의 실수가 올 수 있는데 값이 작을수록 첫번째 파라미터로 오는 이미지의 비중이 많아지고, 값이 커질수록 두번째 파라미터로 오는 이미지의 비중이 많아진다.  imshow("image", blending); // blending 연산한 이미지를 출력한다. 이때 첫번째 파라미터는 이미지를 출력하기 위해 띄우는 창의 제목표시줄에 출력할 제목이고, 두 번째 파라미터는 출력하기 위한 이미지를 가리키는 변수이다.  waitKey(); //키 입력이 있을 때까지 창을 유지하기 위해 waitkey() 함수를 이용한다.  return 0;  } | | |

* + 1. Input : lenna.jpg, orange.jpg

|  |  |  |
| --- | --- | --- |
| Input | | Output |
|  |  |  |
| code | | |
| #include <opencv2/opencv.hpp>  #include "Pixel.h"  using namespace cv;  int main()  {  CPixel processing; //pixel processing을 위한 클래스를 생성한다.  Mat image1 = imread("lenna.png", IMREAD\_COLOR); //lenna.png 를 불러온다. 이때 IMREAD\_COLOR 파라미터는 컬러 이미지를 불러온다는 것을 알려주기 위한 플래그 역할을 한다.  Mat image2 = imread("orange.jpg", IMREAD\_COLOR); // orange.jpg 를 불러온다. 이때 IMREAD\_COLOR 파라미터는 컬러 이미지를 불러온다는 것을 알려주기 위한 플래그 역할을 한다.  //추가로 imread 함수 두 번째 파라미터에는 IMREAD\_COLOR 외에 gray-scale로 읽어들이는 IMREAD\_GRAYSCALE과 alpha channel까지 포함하여 읽어들이는 IMREAD\_UNCHANGED가 올 수 있다.  Mat blending = processing.GS\_blending\_effect(image1, image2, 0.5); // 두 이미지를 이용해 blending 연산을 행한다. 이때 3번째 파라미터에는 0과 1 사이의 실수가 올 수 있는데 값이 작을수록 첫번째 파라미터로 오는 이미지의 비중이 많아지고, 값이 커질수록 두번째 파라미터로 오는 이미지의 비중이 많아진다  imshow("image", blending); // blending 연산한 이미지를 출력한다. 이때 첫번째 파라미터는 이미지를 출력하기 위해 띄우는 창의 제목표시줄에 출력할 제목이고, 두 번째 파라미터는 출력하기 위한 이미지를 가리키는 변수이다.  waitKey(); //키 입력이 있을 때까지 창을 유지하기 위해 waitkey() 함수를 이용한다.  return 0;  } | | |

1. Contrast & brightness adjustment by LUT (Input : 8 bit gray-scale image)
   * 1. Input : stuff\_color\_1.jpg

|  |  |
| --- | --- |
| Input | Output |
|  |  |
| code | |
| #include <opencv2/opencv.hpp>  #include "Pixel.h"  using namespace cv;  int main()  {  CPixel processing; //pixel processing을 위한 클래스를 생성한다.  Mat convert, adjust; //input image를 8bit gray-scale로 변환하고 저장하기 위한 변수와 contrast & brightness adjustment를 행하고 저장하기 위한 변수를 선언한다.  Mat image = imread("stuff\_color\_1.png", IMREAD\_COLOR); // stuff\_color\_1.png 를 불러온다. 이때 IMREAD\_COLOR 파라미터는 컬러 이미지를 불러온다는 것을 알려주기 위한 플래그 역할을 한다.  //추가로 imread 함수 두 번째 파라미터에는 IMREAD\_COLOR 외에 gray-scale로 읽어들이는 IMREAD\_GRAYSCALE과 alpha channel까지 포함하여 읽어들이는 IMREAD\_UNCHANGED가 올 수 있다.  cvtColor(image, convert, CV\_BGR2GRAY); //image 변수에 저장된 이미지를 convert 변수에 8bit gray-scale로 변환하여 저장한다. 이때 3번째 파라미터에는 HSV로 변환하는 COLOR\_BGR2HSV, YCrCb로 변환하는 COLOR\_BGR2YCrCb, Luv로 변환하는 COLOR\_BGR2Luv 등등이 올 수 있다.  adjust = processing.GS\_LUT\_basic\_contrast\_brightness(convert, 0.5, 50); //LUT를 이용한 contrast & brightness adjustment를 행한다. 이때 두번째 파라미터에는 실수 형태의 값이 오는데 값이 커질수록 대비가 증가하여 색의 구분이 명확해지고, 세번째 파라미터에는 정수 값이 오고, 값이 커질수록 밝기가 커진다.  imshow("image", adjust); // adjust에 저장된 이미지를 출력한다. 이때 첫번째 파라미터는 이미지를 출력하기 위해 띄우는 창의 제목표시줄에 출력할 제목이고, 두 번째 파라미터는 출력하기 위한 이미지를 가리키는 변수이다.  waitKey(); //키 입력이 있을 때까지 창을 유지하기 위해 waitkey() 함수를 이용한다.  return 0;  } | |

* + 1. Input : lenna.jpg

|  |  |
| --- | --- |
| Input | Output |
|  |  |
| code | |
| #include <opencv2/opencv.hpp>  #include "Pixel.h"  using namespace cv;  int main()  {  CPixel processing; //pixel processing을 위한 클래스를 생성한다.  Mat convert, adjust; //input image를 8bit gray-scale로 변환하고 저장하기 위한 변수와 contrast & brightness adjustment를 행하고 저장하기 위한 변수를 선언한다.  Mat image = imread("lenna.png", IMREAD\_COLOR); // lenna.png 를 불러온다. 이때 IMREAD\_COLOR 파라미터는 컬러 이미지를 불러온다는 것을 알려주기 위한 플래그 역할을 한다.  cvtColor(image, convert, CV\_BGR2GRAY); //image 변수에 저장된 이미지를 convert 변수에 8bit gray-scale로 변환하여 저장한다. 이때 3번째 파라미터에는 HSV로 변환하는 COLOR\_BGR2HSV, YCrCb로 변환하는 COLOR\_BGR2YCrCb, Luv로 변환하는 COLOR\_BGR2Luv 등등이 올 수 있다.  adjust = processing.GS\_LUT\_basic\_contrast\_brightness(convert, 0.5, 50); //LUT를 이용한 contrast & brightness adjustment를 행한다. 이때 두번째 파라미터에는 실수 형태의 값이 오는데 값이 커질수록 대비가 증가하여 색의 구분이 명확해지고, 세번째 파라미터에는 정수 값이 오고, 값이 커질수록 밝기가 커진다.  imshow("image", adjust); // adjust에 저장된 이미지를 출력한다. 이때 첫번째 파라미터는 이미지를 출력하기 위해 띄우는 창의 제목표시줄에 출력할 제목이고, 두 번째 파라미터는 출력하기 위한 이미지를 가리키는 변수이다.  waitKey(); //키 입력이 있을 때까지 창을 유지하기 위해 waitkey() 함수를 이용한다.  return 0;  } | |