

영상처리 실제 5주차 실습_화소처리

2023254015 장욱진

```
#include <opencv2/opencv.hpp>
#include <stdlib.h>
#include <stdio.h>

using namespace std;
using namespace cv;

Mat src, src_gray, dst;
int threshold_value = 0;
int threshold_type = 0;

void Threshold_Demo(int, void*)
{
    threshold(src_gray, dst, threshold_value, 255, threshold_type);
    imshow("결과 영상", dst);
}

void brighten(Mat& img, int value)
{
    for (int r = 0; r < img.rows; r++)
    {
        for (int c = 0; c < img.cols; ++c)
        {
            img.at<uchar>(r, c) =
                saturate_cast<uchar>(img.at<uchar>(r, c) + value);
        }
    }
}

void page8()
{
    Mat img = imread("./lenna.jpg", IMREAD_GRAYSCALE);
    imshow("Original Image", img);
    brighten(img, 30);
    imshow("New Image", img);
    waitKey(0);
}

void page10()
{
    Mat img = imread("d:/lenna.jpg", IMREAD_GRAYSCALE);
    imshow("Original Image", img);
    for (int r = 0; r < img.rows; r++) {
        uchar* p = img.ptr<uchar>(r);
        for (int c = 0; c < img.cols; ++c) {
            p[c] = saturate_cast<uchar>(p[c] + 30);
        }
    }
    imshow("New Image", img);
    waitKey(0);
}

void page14()
{
    double alpha = 1.0;
    int beta = 0;
```

```

    Mat image = imread("./contrast.jpg");
    Mat oimage;
    cout << "알파값을 입력하시오: [1.0-3.0]: "; cin >> alpha;
    cout << "베타값을 입력하시오: [0-100]: "; cin >> beta;
    image.convertTo(oimage, -1, alpha, beta);
    imshow("Original Image", image);
    imshow("New Image", oimage);
    waitKey();
}

void page21()
{
    src = imread("./lenna.png");
    cvtColor(src, src_gray, CV_BGR2GRAY);
    namedWindow("결과 영상", CV_WINDOW_AUTOSIZE);

    createTrackbar("임계값", "결과영상", &threshold_value, 255, Threshold_Demo);

    Threshold_Demo(0, 0);

    while (true)
    {
        int c;
        c = waitKey(20);
        if ((char)c == 27)
        {
            break;
        }
    }
}

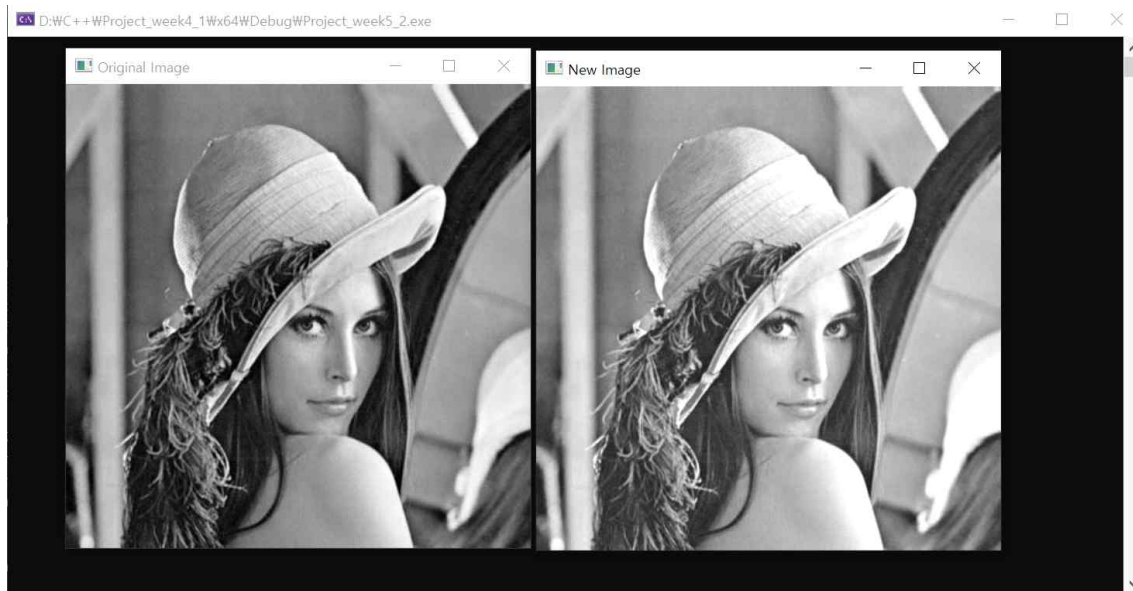
int page31()
{
    Mat src1, src2, dst;
    double gamma = 0.5;
    src1 = imread("./gamma1.jpg");
    if (src1.empty()) { cout << "영상을 읽을 수 없습니다." << endl; return -1; }
    Mat table(1, 256, CV_8U);
    uchar* p = table.ptr();
    for (int i = 0; i < 256; ++i)
        p[i] = saturate_cast<uchar> (pow(i / 255.0, gamma) * 255.0);
    LUT(src1, table, dst);
    imshow("src1", src1);
    imshow("dst", dst);
    waitKey(0);
}

int main()
{
    page8();
    page10();
    page14();
    page21();
    page31();

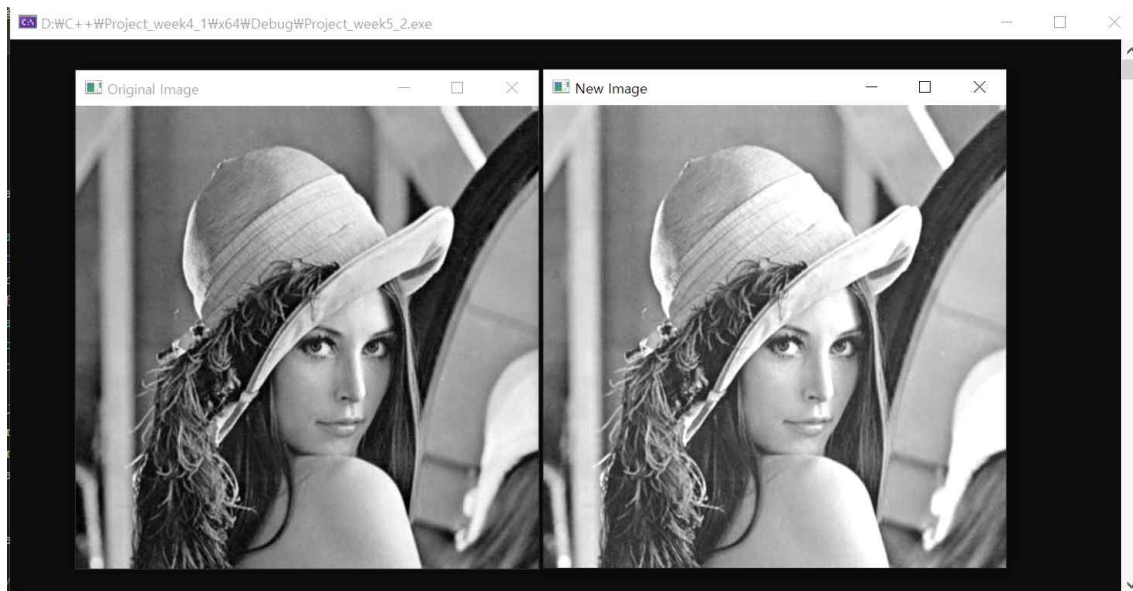
    return 0;
}

```

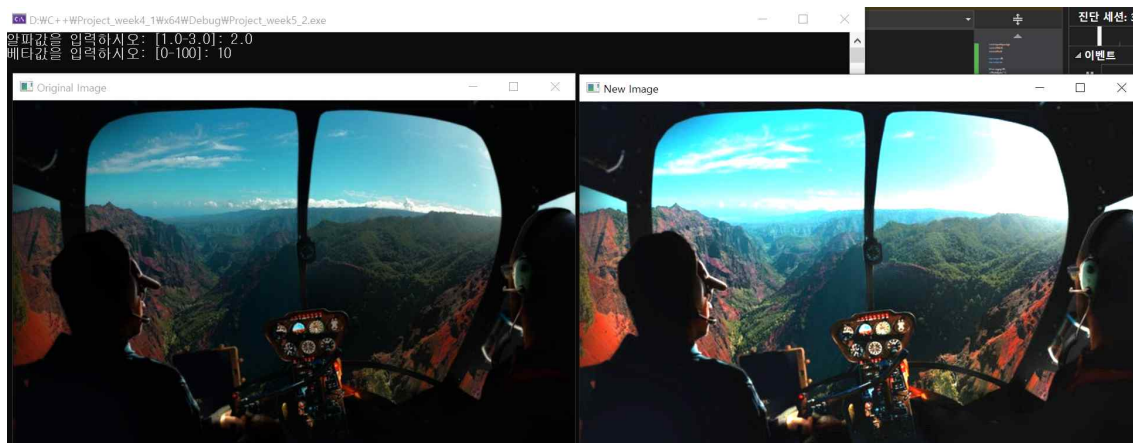
결과화면



<page8 결과화면>



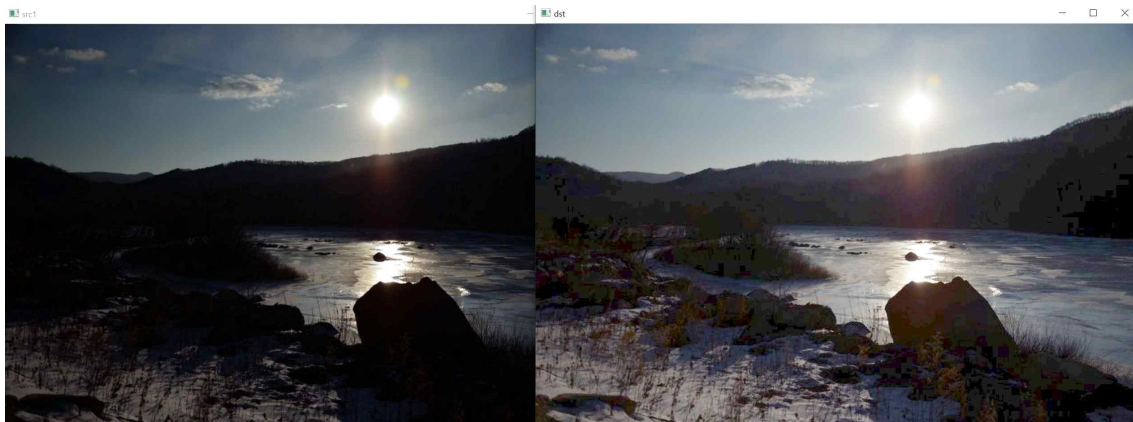
<page10 결과화면>



<page14 결과화면>



<page21 결과화면>



<page31 결과화면>