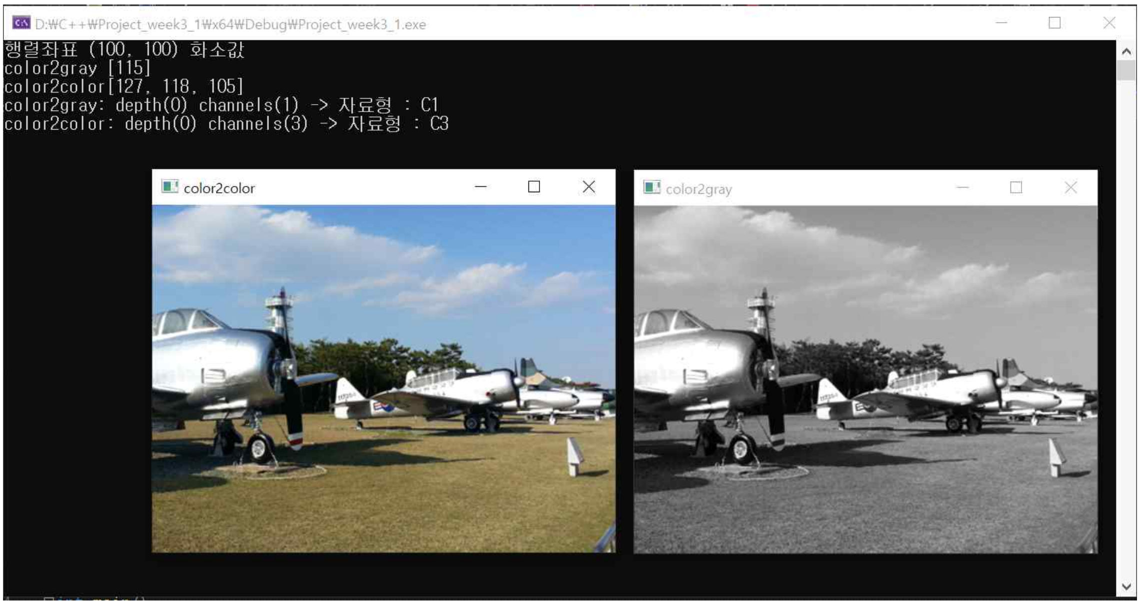
영상처리 실제 3주차 실습(1)

2023254015 장욱진

```
#include <opencv2/opencv.hpp>

using namespace std;
using namespace cv;

void print_matInfo(string name, Mat img)
{
	string mat_type;
	if (img.depth() == CV_8U) mat_type == "CV_8U";
	else if (img.depth() == CV_8S) mat_type == "CV_8S";
	else if (img.depth() == CV_16U) mat_type == "CV_16U";
	else if (img.depth() == CV_32S) mat_type == "CV_16S";
	else if (img.depth() == CV_32F) mat_type == "CV_32F";
	else if (img.depth() == CV_64F) mat_type == "CV_64F";

	cout << name;
	cout << format(": depth(%d) channels(%d) -> 자료형 : ", img.depth(), img.channels());
	cout << mat_type << "C" << img.channels() << endl;
}

void put_string(Mat frame, string text, Point pt, int value)
{
	text += to_string(value);
	Point shade = pt + Point(2, 2);
	int font = FONT_HERSHEY_SIMPLEX;
	putText(frame, text, shade, font, 0.7, Scalar(0, 0, 0), 2);
	putText(frame, text, pt, font, 0.7, Scalar(120, 200, 90), 2);
}
VideoCapture capture_global;

void zoom_bar(int value, void*)
{
	capture_global.set(CAP_PROP_ZOOM, value);
}

void focus_bar(int value, void*)
{
	capture_global.set(CAP_PROP_FOCUS, value);
}

void page26()
{
	VideoCapture cap("./trailer.mp4");
	if (!cap.isOpened())
```

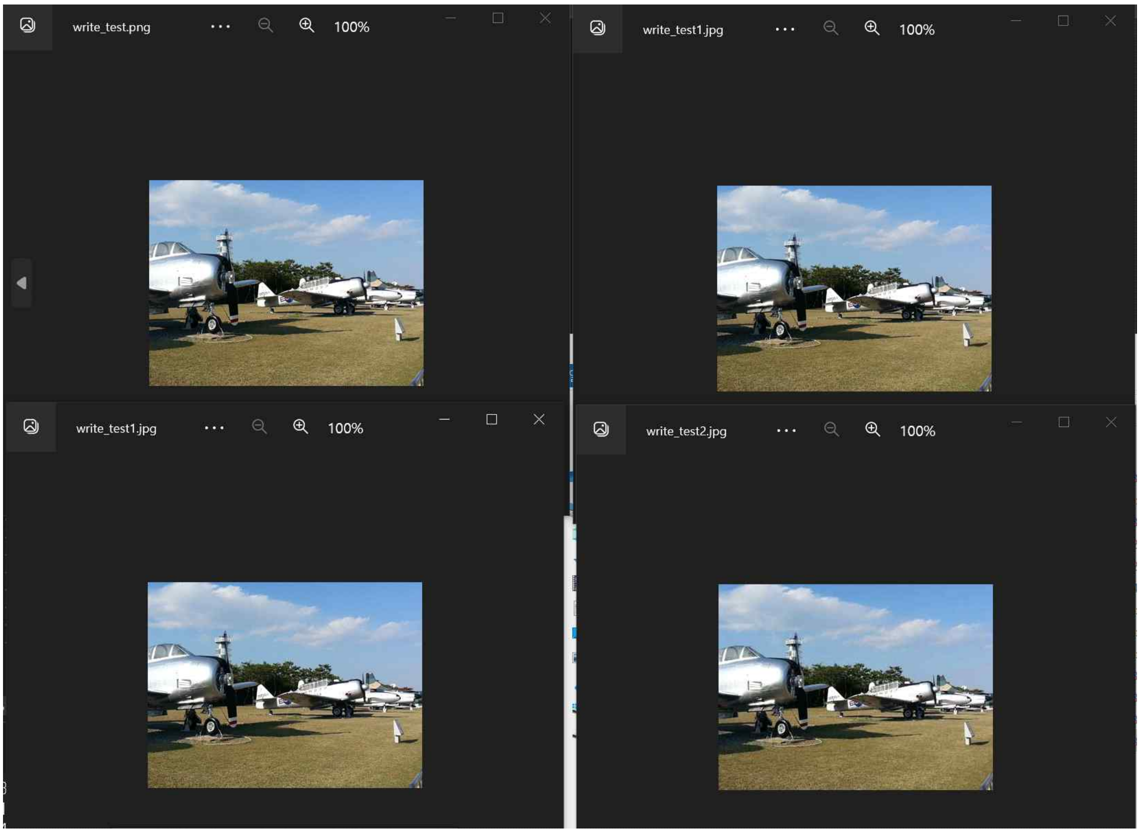
```
    {
        cout << "동영상을 읽을 수 없음" << endl;
    }

    namedWindow("frame", 1); //윈도우 생성
    for (;;)
    {
        Mat frame;
        cap >> frame; // 동영상에서 하나의 프레임을 추출한다.
        imshow("frame", frame);
        if (waitKey(30) >= 0)break;
    }
}

void page23()
{
    VideoCapture capture;
    capture.open("./video_file.avi");
    CV_Assert(capture.isOpened());

    double frame_rate = capture.get(CV_CAP_PROP_FPS);
    int delay = 1000 / frame_rate;
    int frame_cnt = 0;
    Mat frame;

    while (capture.read(frame))
    {
        if (waitKey(delay) >= 0)break;

        if (frame_cnt < 100);
        else if (frame_cnt < 200) frame -= Scalar(0, 0, 100);
        else if (frame_cnt < 300) frame += Scalar(100, 0, 0);
        else if (frame_cnt < 400) frame = frame * 1.5;
        else if (frame_cnt < 500) frame = frame * 0.5;

        put_string(frame, "frame_cnt ", Point(20, 50), frame_cnt);
        imshow("동영상 파일읽기", frame);
    }


}

void page20()
{
    VideoCapture capture(0);
    CV_Assert(capture.isOpened());

    double fps = 29.97;
    int delay = cvRound(1000.0 / fps);
    Size size(640, 360);
    int fourcc = VideoWriter::fourcc('D', 'X', '5', '0');

    capture.set(CAP_PROP_FRAME_WIDTH, size.width);
    capture.set(CAP_PROP_FRAME_HEIGHT, size.height);

    cout << "width x height : " << size << endl;
    cout << "VideoWriter::fourcc : " << fourcc << endl;
    cout << "delay : " << delay << endl;
    cout << "fps : " << fps << endl;

    VideoWriter writer;
    writer.open("./video_file.avi", fourcc, fps, size);
    CV_Assert(writer.isOpened());

    for (;;)
    {
        Mat frame;
        capture >> frame;
        writer << frame;

        imshow("카메라 영상보기", frame);
        if (waitKey(delay) >= 0)
            break;

    }
}

void page17()
{
    capture_global.open(0);
    CV_Assert(capture_global.isOpened());

    capture_global.set(CAP_PROP_FRAME_WIDTH, 400);
    capture_global.set(CAP_PROP_FRAME_HEIGHT, 300);
    capture_global.set(CAP_PROP_AUTOFOCUS, 0);
    capture_global.set(CAP_PROP_BRIGHTNESS, 150);

    int zoom = capture_global.get(CAP_PROP_ZOOM);
    int focus = capture_global.get(CAP_PROP_FOCUS);

    string title = "카메라 속성변경";
    namedWindow(title);
    createTrackbar("zoom", title, &zoom, 10, zoom_bar);
    createTrackbar("focus", title, &focus, 40, focus_bar);

    for (;;)
    {
        Mat frame;
        capture_global >> frame;
        put_string(frame, "zoom : ", Point(10, 240), zoom);
        put_string(frame, "focus : ", Point(10, 270), focus);

        imshow(title, frame);
        if (waitKey(30) >= 0)break;
    }
}

void page4()
{
    string filename = "read_color.jpg";
    Mat color2gray = imread(filename, IMREAD_GRAYSCALE);
    Mat color2color = imread(filename, IMREAD_COLOR);
    CV_Assert(color2gray.data && color2color.data);


    Rect roi(100, 100, 1, 1);
    cout << "행렬좌표 (100, 100) 화소값 " << endl;
    cout << "color2gray " << color2gray(roi) << endl;
    cout << "color2color" << color2color(roi) << endl;

    print_matInfo("color2gray", color2gray);
    print_matInfo("color2color", color2color);
    imshow("color2gray", color2gray);
    imshow("color2color", color2color);
    waitKey(0);
}

void page15()
{
    VideoCapture capture1(0);
    if (!capture1.isOpened())
    {
        cout << "카메라가 연결되지 않았습니다." << endl;
        exit(1);
    }

    cout << "너비 " << capture1.get(CAP_PROP_FRAME_WIDTH) << endl;
    cout << "높이 " << capture1.get(CAP_PROP_FRAME_HEIGHT) << endl;
    cout << "노출 " << capture1.get(CAP_PROP_EXPOSURE) << endl;
    cout << "밝기 " << capture1.get(CAP_PROP_BRIGHTNESS) << endl;

    for (;;)
    {
        Mat frame;
        capture1.read(frame);

        put_string(frame, "EXPOS : ", Point(10, 40),
capture1.get(CAP_PROP_EXPOSURE));
        imshow("카메라 영상보기", frame);
        if (waitKey(30) >= 0)break;
    }
}

void page8()
{
    Mat img8 = imread("read_color.jpg", IMREAD_COLOR);
    CV_Assert(img8.data);

    vector<int> params_jpg, params_png;
    params_jpg.push_back(IMWRITE_JPEG_QUALITY);
    params_jpg.push_back(50);
    params_png.push_back(IMWRITE_PNG_COMPRESSION);
    params_png.push_back(9);

    imwrite("write_test1.jpg", img8);
    imwrite("write_test2.jpg", img8, params_jpg);
    imwrite("write_test.png", img8, params_png);
    imwrite("write_test.bmp", img8);
}

int main()
{
    page4();
    page8();
    page15();
    page17();
    page20();
    page23();
    page26();

    return 0;
}
```

결과화면

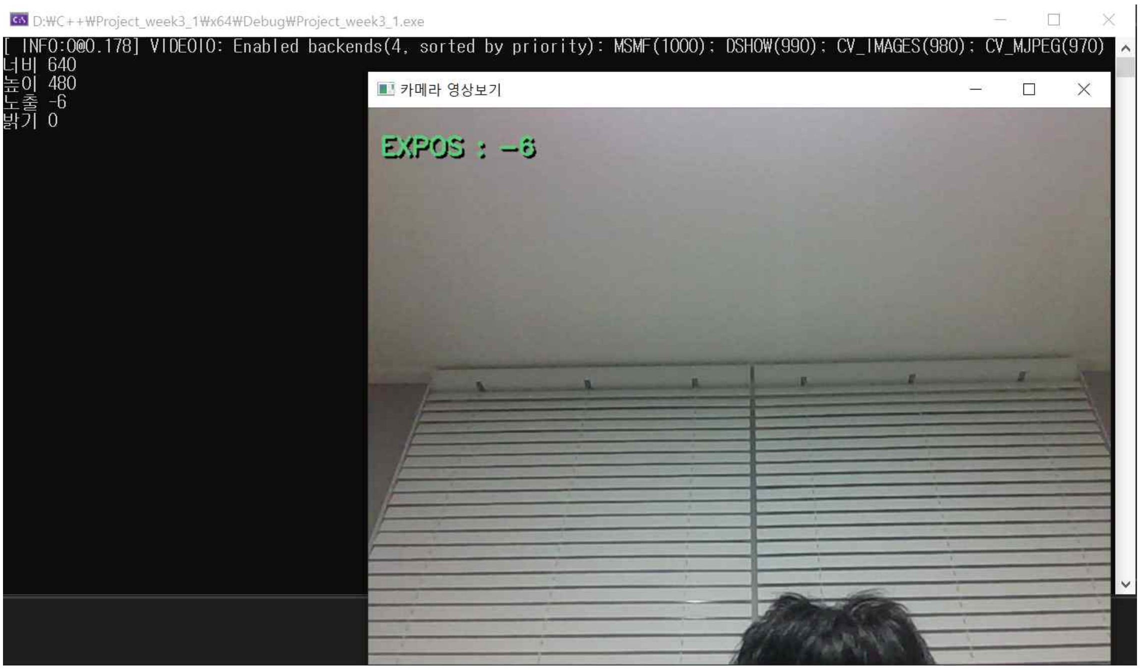


```
행렬좌표 (100, 100) 화소값
color2gray [115]
color2color[127, 118, 105]
color2gray: depth(0) channels(1) -> 자료형 : C1
color2color: depth(0) channels(3) -> 자료형 : C3
```

<page4 결과화면>



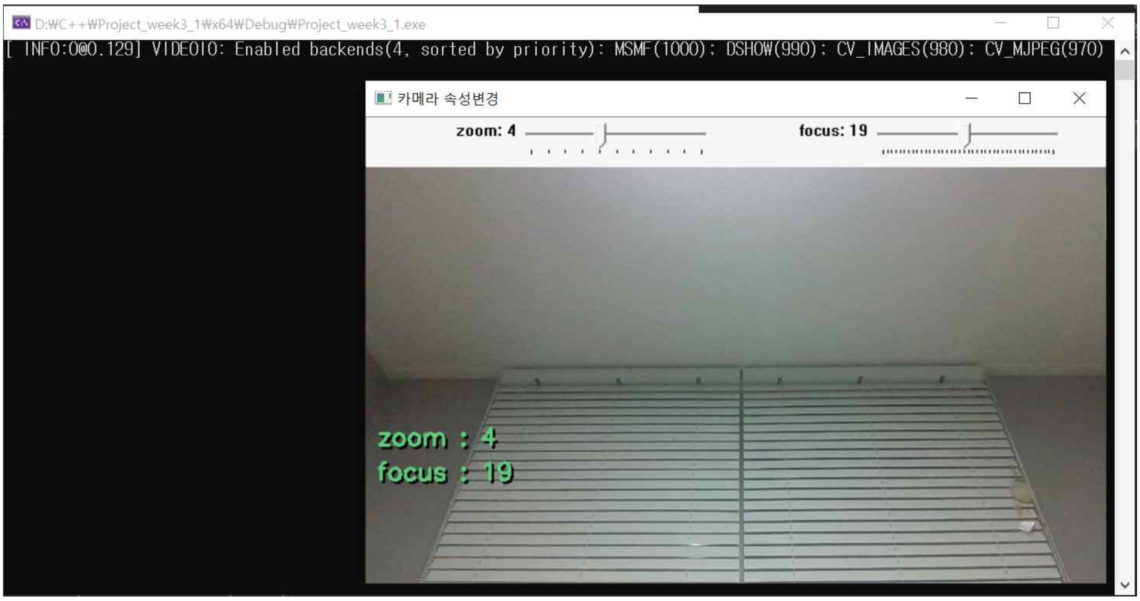
<page8 결과화면>



D:₩C++₩Project_week3_1₩x64₩Debug₩Project_week3_1.exe

```
[ INFO:0@0.178] VIDEOIO: Enabled backends(4, sorted by priority): MSMF(1000); DSHOW(990); CV_IMAGES(980); CV_MJPEG(970)
너비 640
높이 480
노출 -6
밝기 0
```

<page15 결과화면>



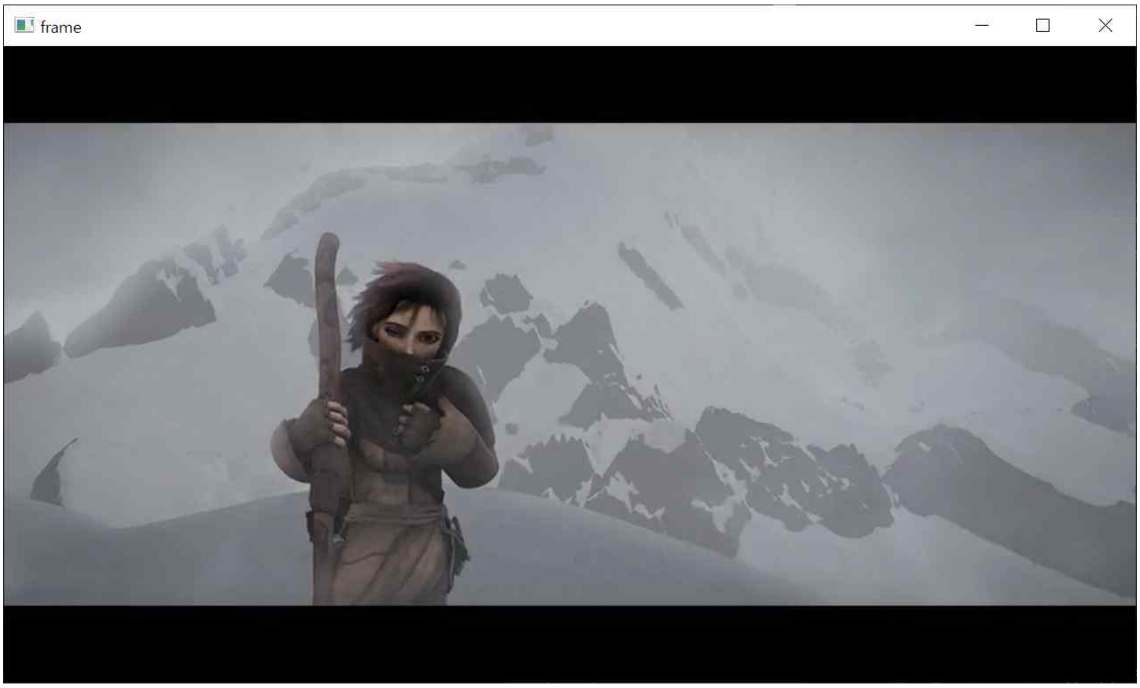
D:₩C++₩Project_week3_1₩x64₩Debug₩Project_week3_1.exe

```
[ INFO:0@0.129] VIDEOIO: Enabled backends(4, sorted by priority): MSMF(1000); DSHOW(990); CV_IMAGES(980); CV_MJPEG(970)
```

zoom: 4
focus: 19

<page17 결과화면>

<page20, 23 실행실패>


frame

<page26 결과화면>