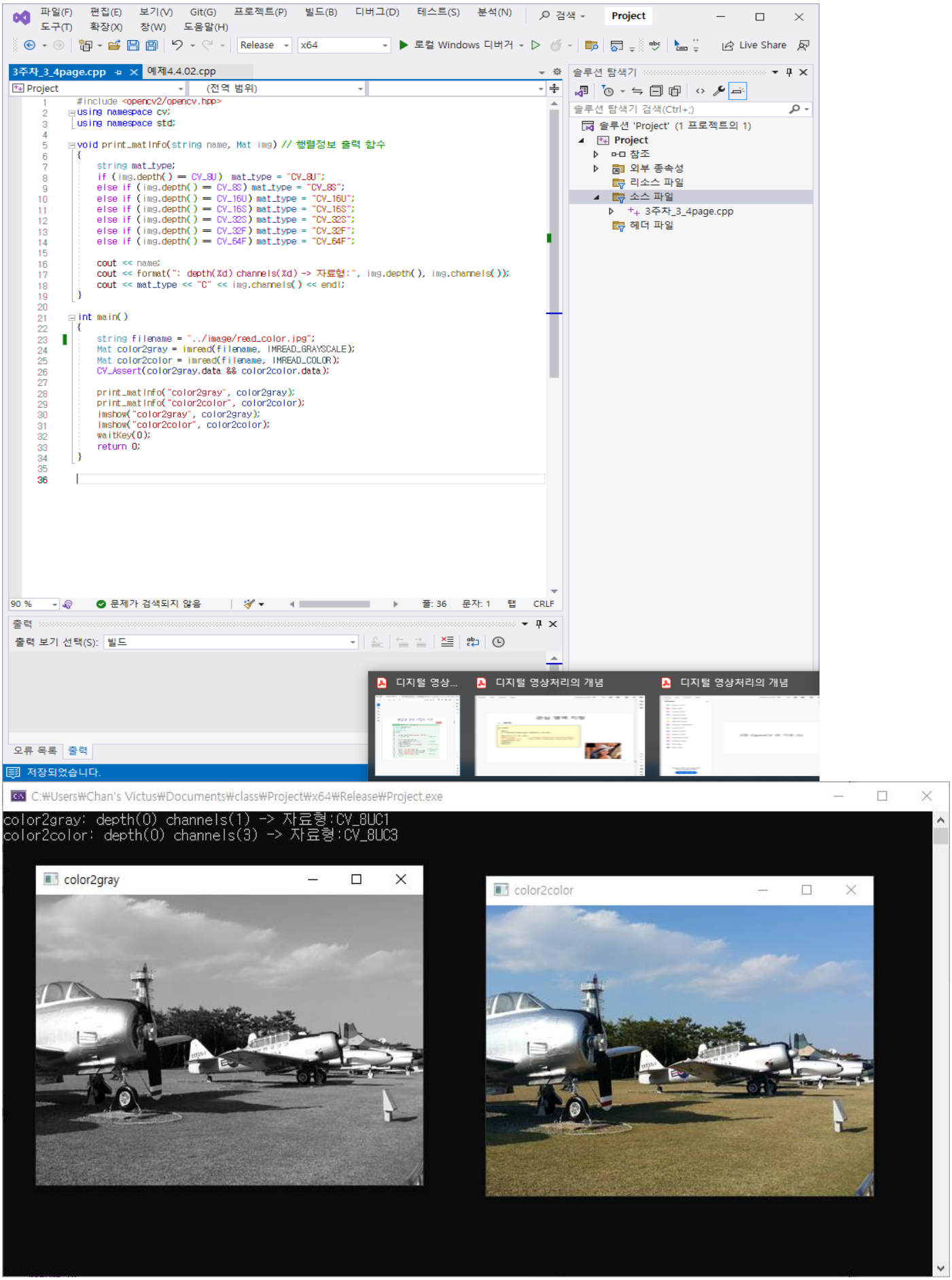
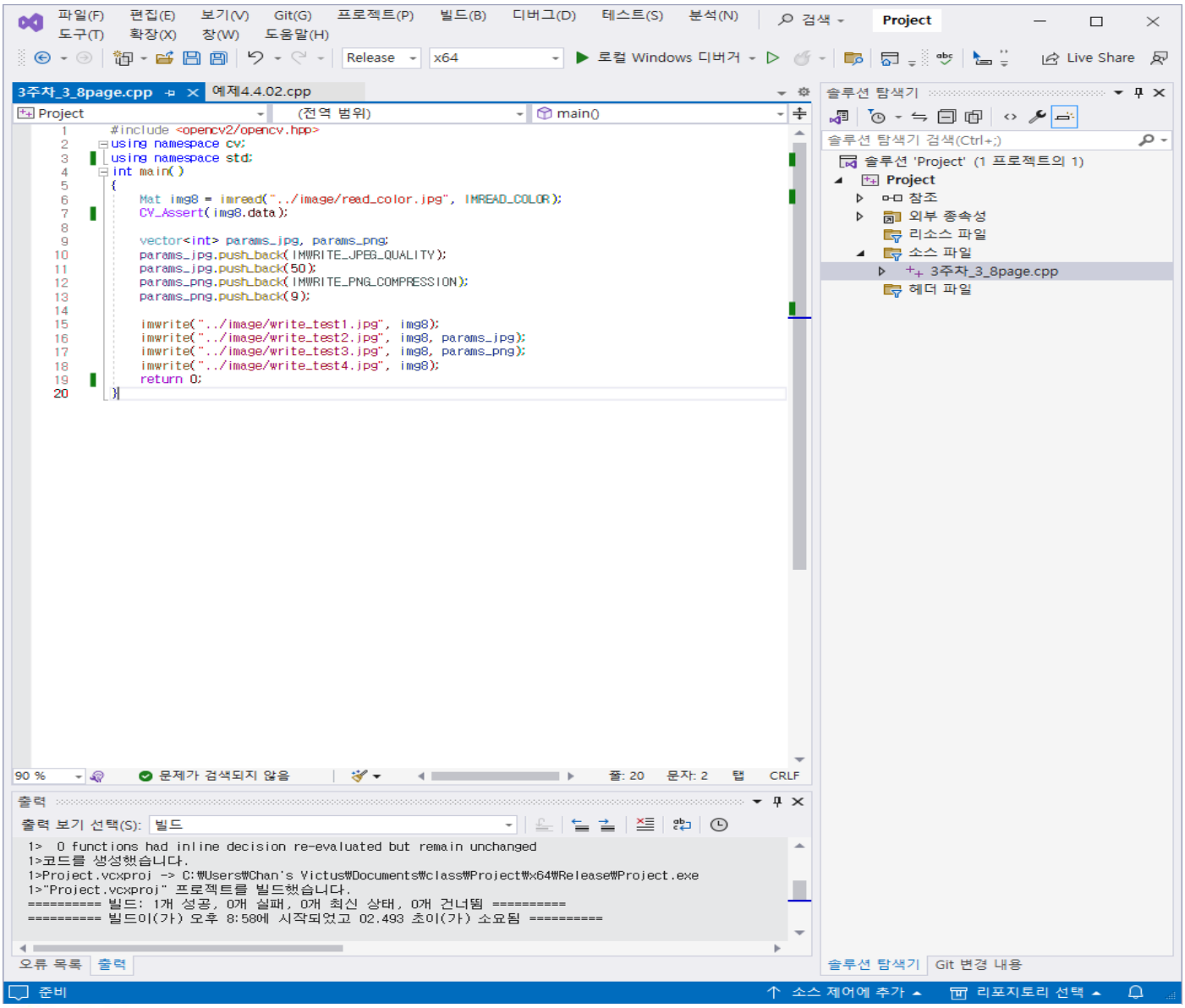
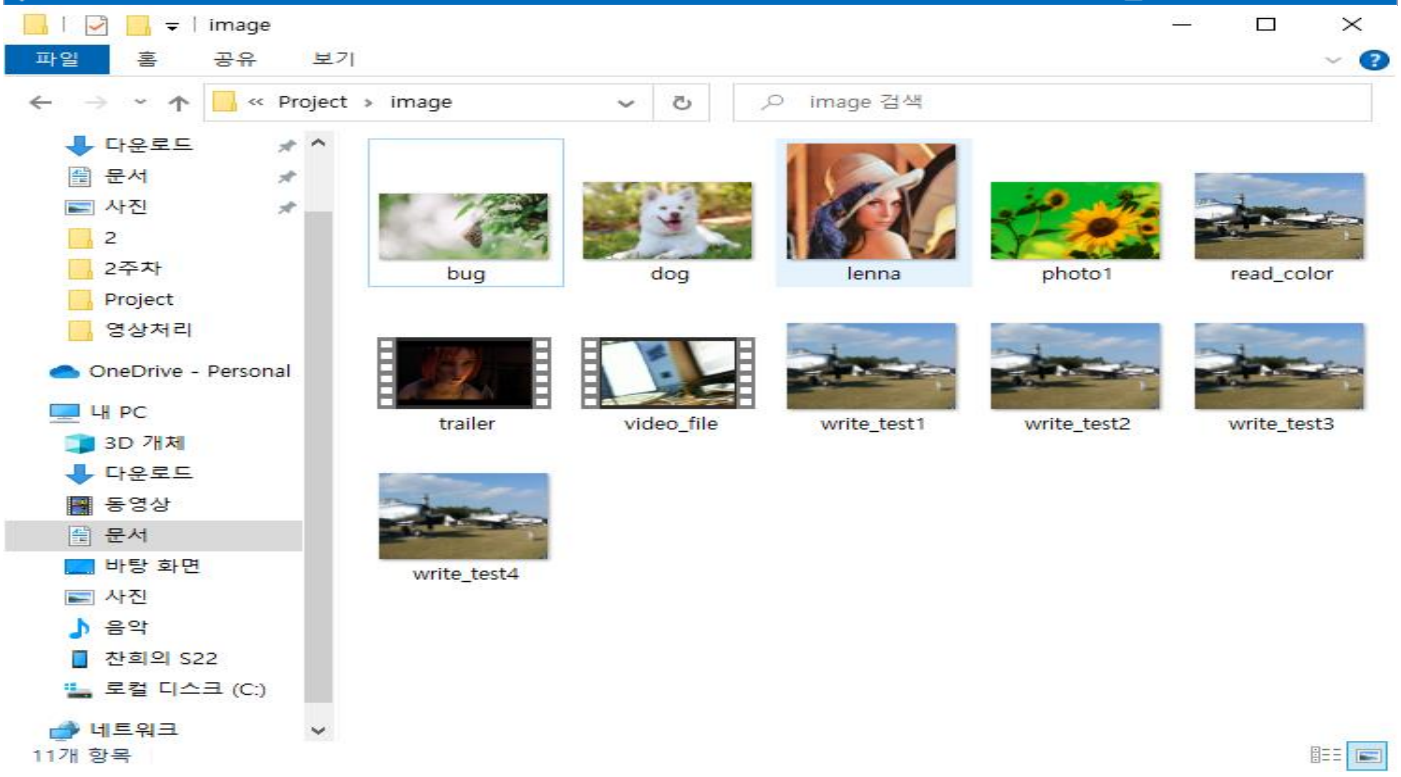
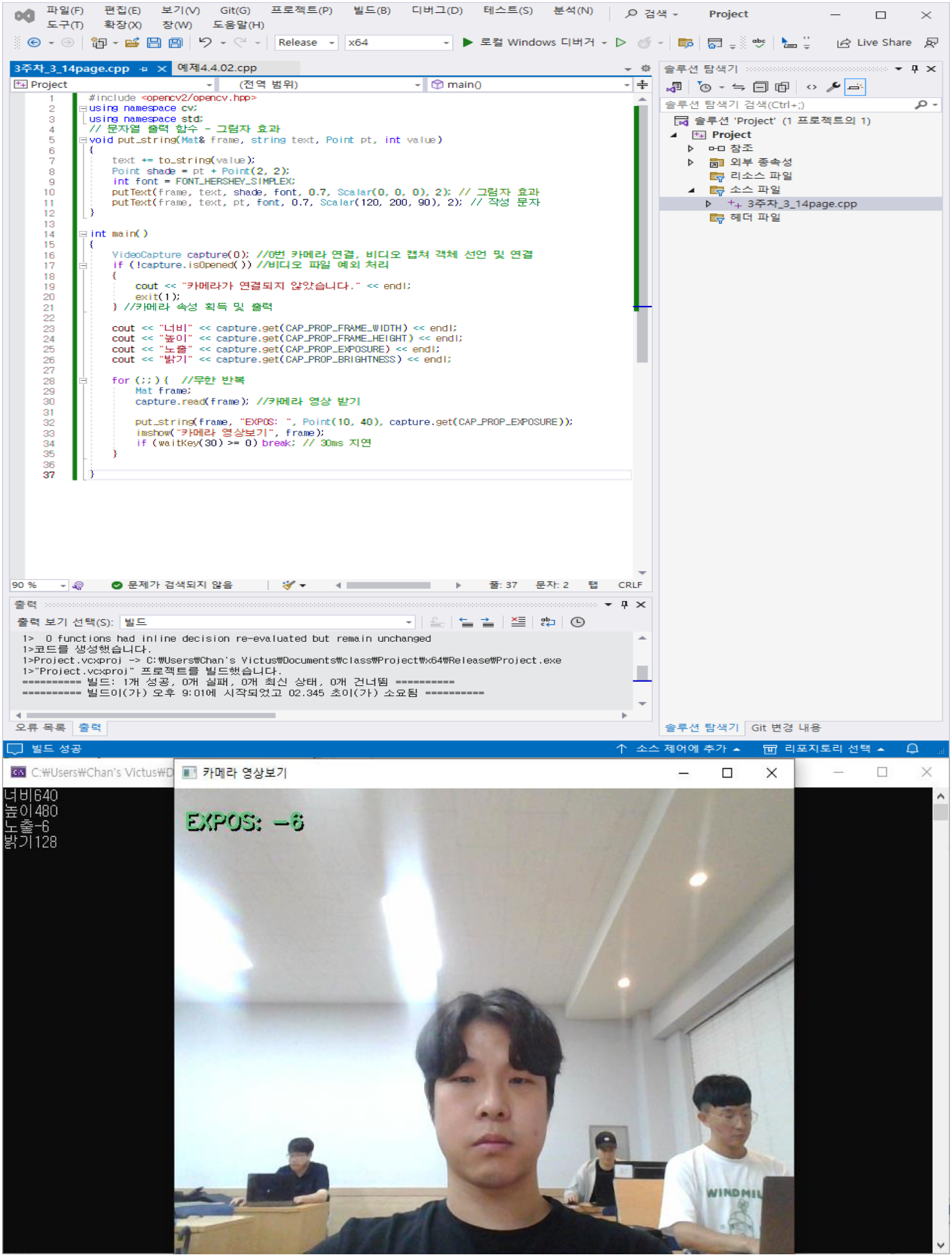
03-OpenCV의 기초(3) 4page



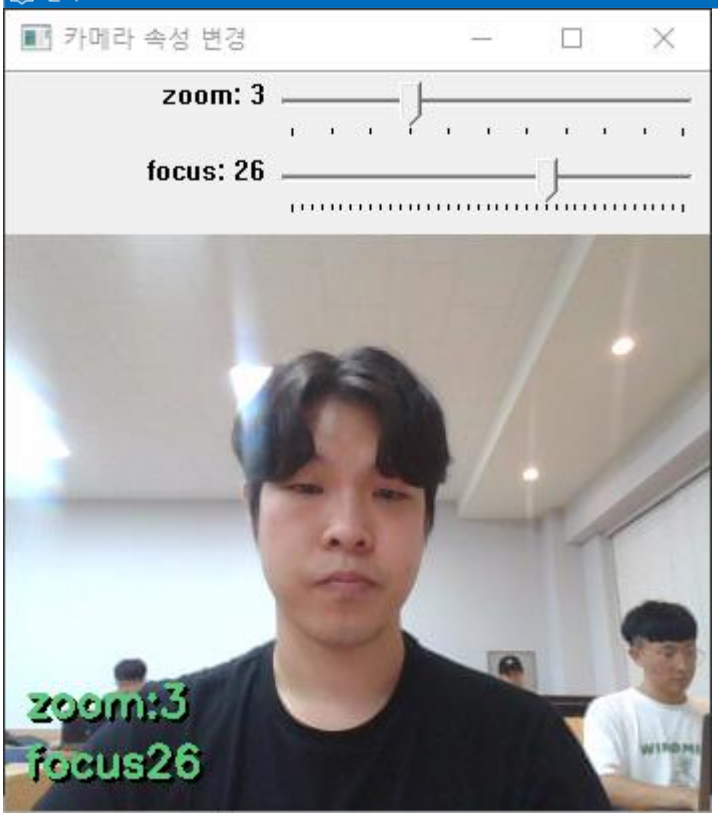
03-OpenCV의 기초(3) 8page

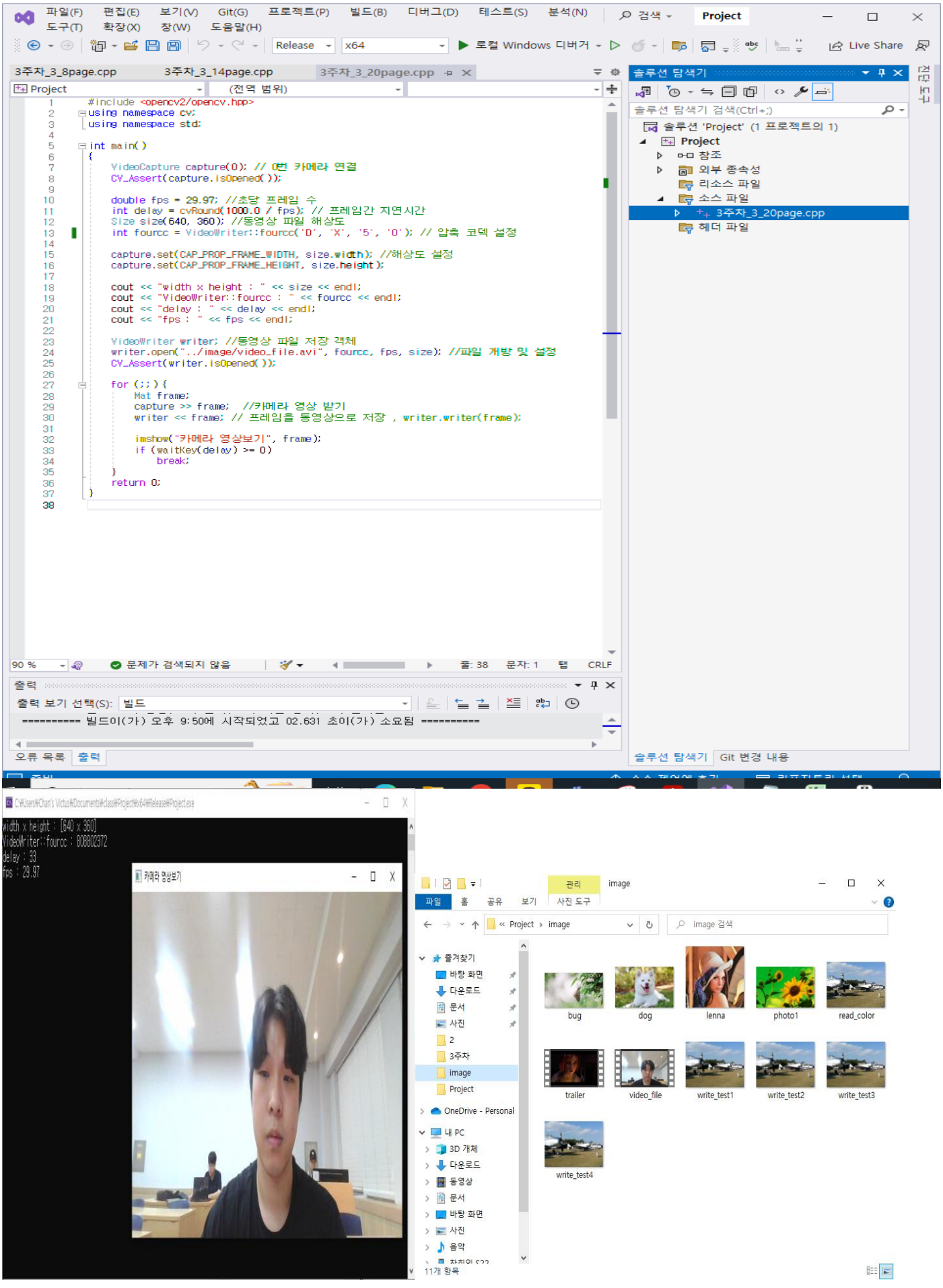




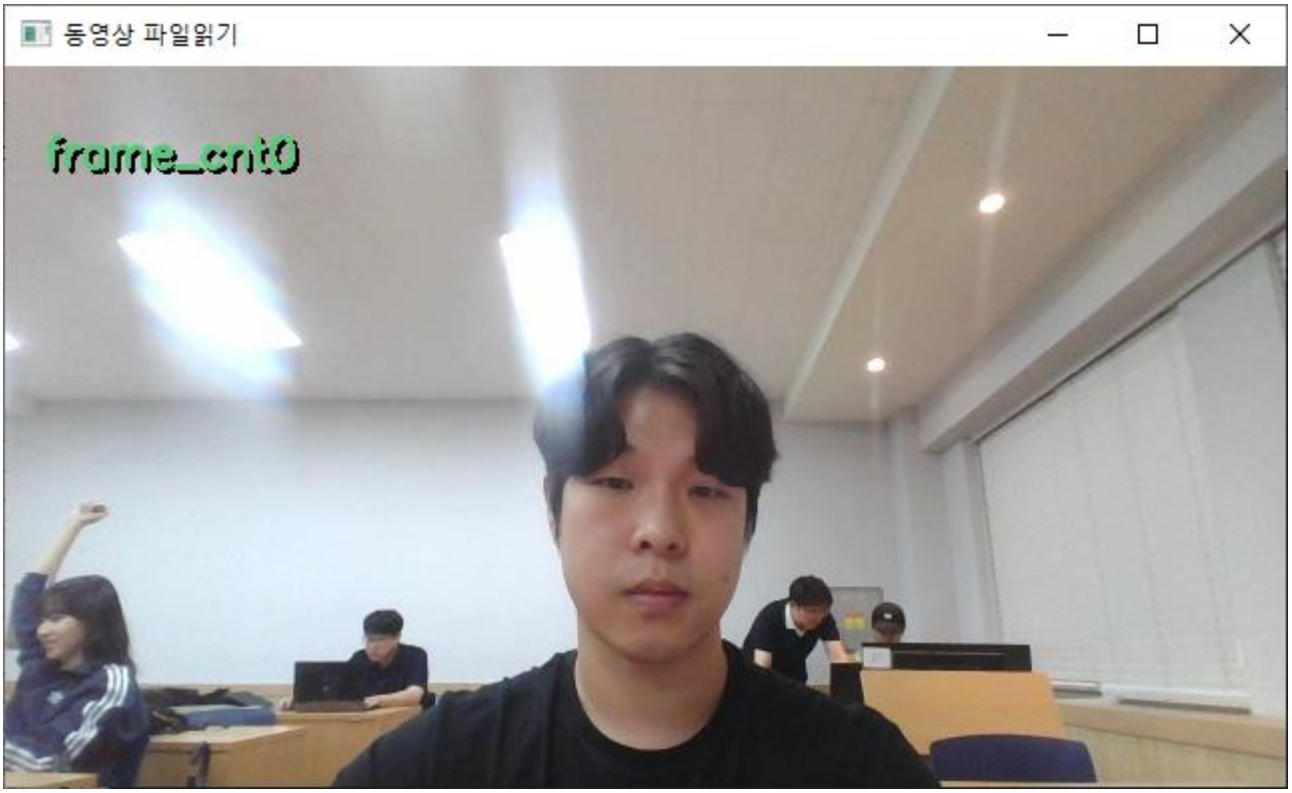

파일(F) 편집(E) 보기(V) Git(G) 프로젝트(P) 빌드(B) 디버그(D) 테스트(S) 분석(N)
도구(T) 확장(X) 창(W) 도움말(H)
검색
Project
Release
x64
로컬 Windows 디버거
Live Share
3주차_3_14page.cpp
예제4.4.02.cpp
Project
(전역 범위)
main()
#include <opencv2/opencv.hpp>
using namespace cv;
using namespace std;
// 문자열 출력 함수 - 그림자 효과
void put_string(Mat& frame, 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); // 그림자 효과
putText(frame, text, pt, font, 0.7, Scalar(120, 200, 90), 2); // 작성 문자
}
int main()
{
VideoCapture capture(0); //0번 카메라 연결, 비디오 캡쳐 객체 선언 및 연결
if (!capture.isOpened()) //비디오 파일 예외 처리
{
cout << "카메라가 연결되지 않았습니다." << endl;
exit(1);
} //카메라 속성 획득 및 출력
cout << "너비" << capture.get(CAP_PROP_FRAME_WIDTH) << endl;
cout << "높이" << capture.get(CAP_PROP_FRAME_HEIGHT) << endl;
cout << "노출" << capture.get(CAP_PROP_EXPOSURE) << endl;
cout << "밝기" << capture.get(CAP_PROP_BRIGHTNESS) << endl;
for (;;) { //무한 반복
Mat frame;
capture.read(frame); //카메라 영상 받기
put_string(frame, "EXPOS: ", Point(10, 40), capture.get(CAP_PROP_EXPOSURE));
imshow("카메라 영상보기", frame);
if (waitKey(30) >= 0) break; // 30ms 지연
}
}
90 %
문제가 검색되지 않음
줄: 37 문자: 2 탭 CRLF
솔루션 탐색기
솔루션 탐색기 검색(Ctrl+;)
솔루션 'Project' (1 프로젝트의 1)
Project
참조
외부 종속성
리소스 파일
소스 파일
3주차_3_14page.cpp
헤더 파일
솔루션 탐색기 Git 변경 내용
출력
출력 보기 선택(S): 빌드
1> 0 functions had inline decision re-evaluated but remain unchanged
1>코드를 생성했습니다.
1>Project.vcxproj -> C:₩Users₩Chan's Victus₩Documents₩class₩Project₩x64₩Release₩Project.exe
1>"Project.vcxproj" 프로젝트를 빌드했습니다.
========== 빌드: 1개 성공, 0개 실패, 0개 최신 상태, 0개 건너뜀 ==========
========== 빌드이(가) 오후 9:01에 시작되었고 02.345 초이(가) 소요됨 ==========
오류 목록 출력
빌드 성공
소스 제어에 추가
리포지토리 선택
C:₩Users₩Chan's Victus₩D
너비640
높이480
노출-6
밝기128
카메라 영상보기
EXPOS: -6

```cpp
#include <opencv2/opencv.hpp>
using namespace cv;
using namespace std;

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; // 전역 변수 선언 - 여러 함수에서 사용
void zoom_bar(int value, void*) {
    capture.set(CAP_PROP_ZOOM, value);
} // 트랙바 콜백 함수, 줌 설정
void focus_bar(int value, void*) {
    capture.set(CAP_PROP_FOCUS, value);
} // 초점 설정
int main()
{
    capture.open(0); // 0번 카메라 연결
    CV_Assert(capture.isOpened());

    capture.set(CAP_PROP_FRAME_WIDTH, 400); //카메라 너비
    capture.set(CAP_PROP_FRAME_HEIGHT, 300);
    capture.set(CAP_PROP_AUTOFOCUS, 0); // 오토포커싱 중지
    capture.set(CAP_PROP_BRIGHTNESS, 150); // 프레임 밝기 초기화

    int zoom = capture.get(CAP_PROP_ZOOM); // 카메라 속성 가져오기
    int focus = capture.get(CAP_PROP_FOCUS);

    string title = "카메라 속성 변경"; //윈도우 이름 지정
    namedWindow(title);
    createTrackbar("zoom", title, &zoom, 10, zoom_bar);
    createTrackbar("focus", title, &focus, 40, focus_bar);
    for (;;) {
        Mat frame;
        capture >> frame; //카메라 영상 받기

        put_string(frame, "zoom: ", Point(10, 240), zoom); // 줌 값 영상 표시
        put_string(frame, "focus", Point(10, 270), focus); // 포커스

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



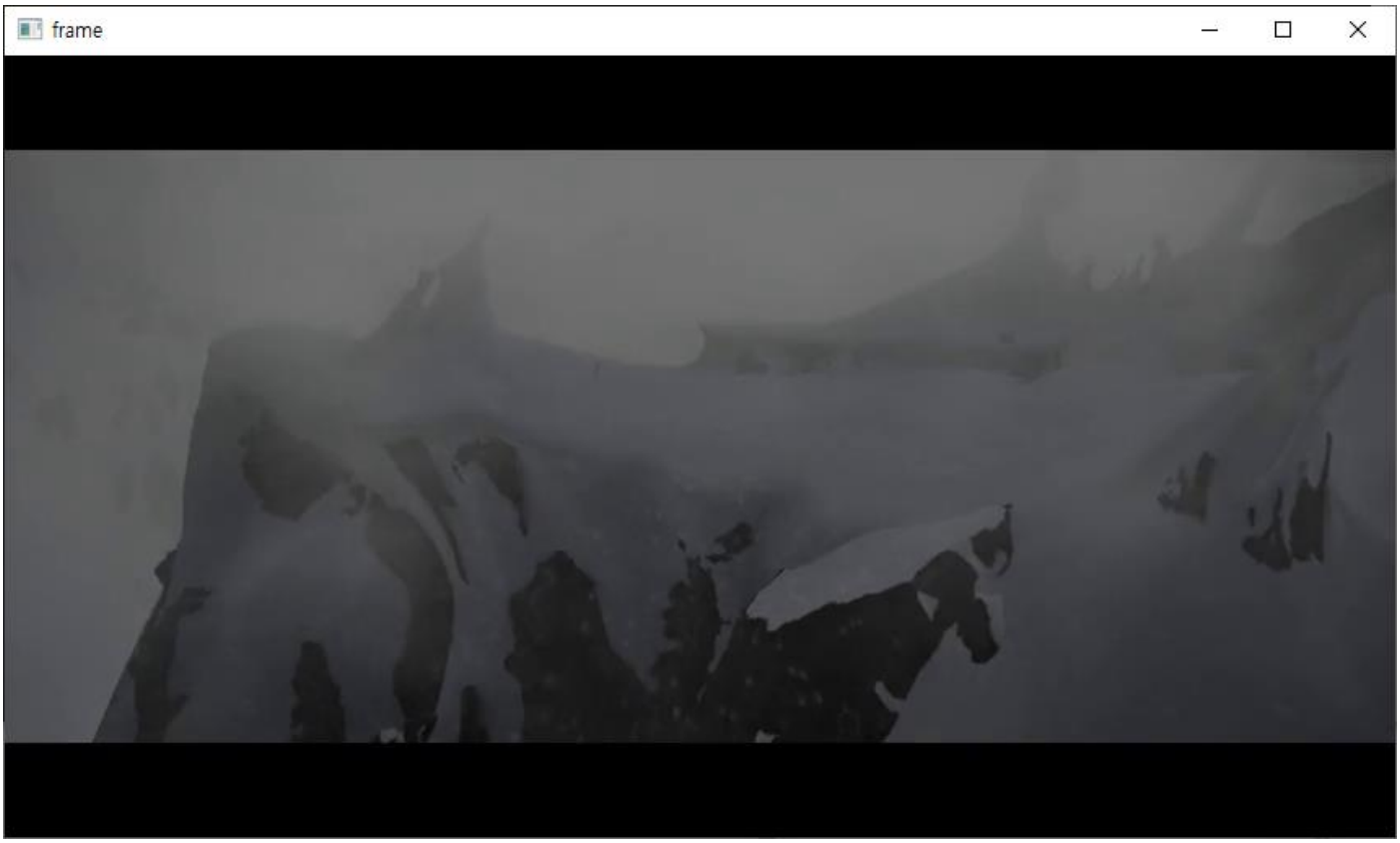

3주차_3_8page.cpp
3주차_3_14page.cpp
3주차_3_20page.cpp
#include <opencv2/opencv.hpp>
using namespace cv;
using namespace std;
int main( )
{
VideoCapture capture(0); // 0번 카메라 연결
CV_Assert(capture.isOpened( ));
double fps = 29.97; //초당 프레임 수
int delay = cvRound(1000.0 / fps); // 프레임간 지연시간
Size size(640, 360); //동영상 파일 해상도
int fourcc = VideoWriter::fourcc('D', 'X', '5', '0'); // 압축 코덱 설정
capture.set(CAP_PROP_FRAME_WIDTH, size.width); //해상도 설정
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; //동영상 파일 저장 객체
writer.open("../image/video_file.avi", fourcc, fps, size); //파일 개방 및 설정
CV_Assert(writer.isOpened( ));
for (;;) {
Mat frame;
capture >> frame; //카메라 영상 받기
writer << frame; // 프레임을 동영상으로 저장 , writer.writer(frame);
imshow("카메라 영상보기", frame);
if (waitKey(delay) >= 0)
break;
}
return 0;
}
솔루션 탐색기
솔루션 'Project' (1 프로젝트의 1)
Project
참조
외부 종속성
리소스 파일
소스 파일
3주차_3_20page.cpp
헤더 파일
출력
빌드이(가) 오후 9:50에 시작되었고 02.631 초이(가) 소요됨
width x height : [640 x 360]
VideoWriter::fourcc : 808802372
delay : 33
fps : 29.97
카메라 영상보기
image
bug
dog
lenna
photo1
read_color
trailer
video_file
write_test1
write_test2
write_test3
write_test4

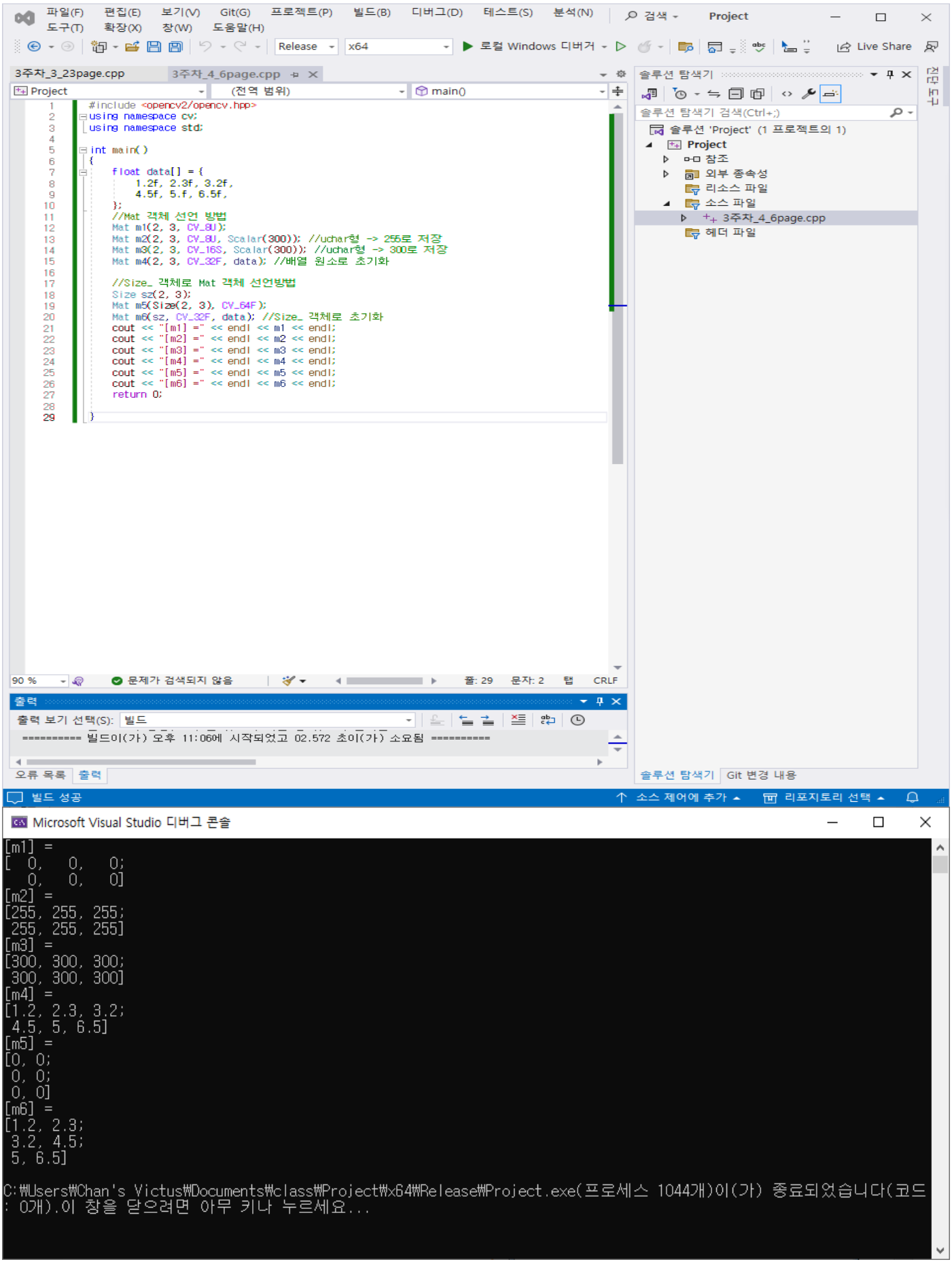
```cpp
#include <opencv2/opencv.hpp>
using namespace cv;
using namespace std;

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);
}
int main( )
{
    VideoCapture capture;
    capture.open("../image/video_file.avi"); // 동영상 파일 개방
    CV_Assert(capture.isOpened( ));

    double frame_rate = capture.get(CAP_PROP_FPS); // 초당 프레임 수
    int delay = 1000 / frame_rate; // 지연시간
    int frame_cnt = 0; //현재 프레임의 번호
    Mat frame;

    while (capture.read(frame)) //프레임 반복 재생
    {
        if (waitKey(delay) >= 0) break; // 프레임간 지연시간 지정

        if (frame_cnt < 100);
        else if (frame_cnt < 200) frame -= Scalar(0, 0, 100); //199프레임까지 빨간색 증가
        else if (frame_cnt < 300) frame += Scalar(100, 0, 0); //299프레임까지 파란색 증가
        else if (frame_cnt < 400) frame = frame * 1.5; //399프레임까지 대비 증가
        else if (frame_cnt < 500) frame = frame * 0.5; //499프레임까지 대비 감소

        put_string(frame, "frame_cnt", Point(20, 50), frame_cnt);
        imshow("동영상 파일읽기", frame);
    }
    return 0;
}
```

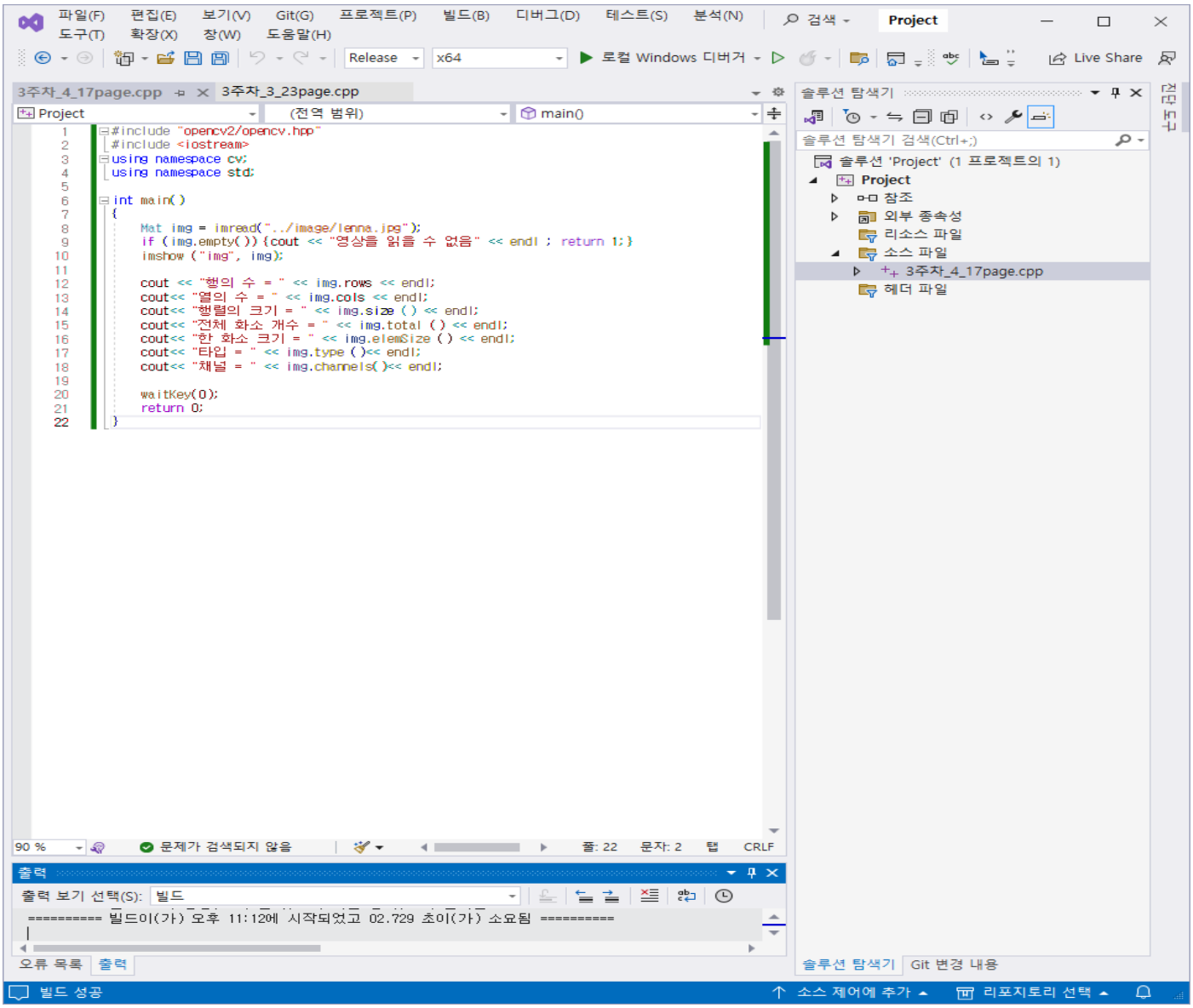


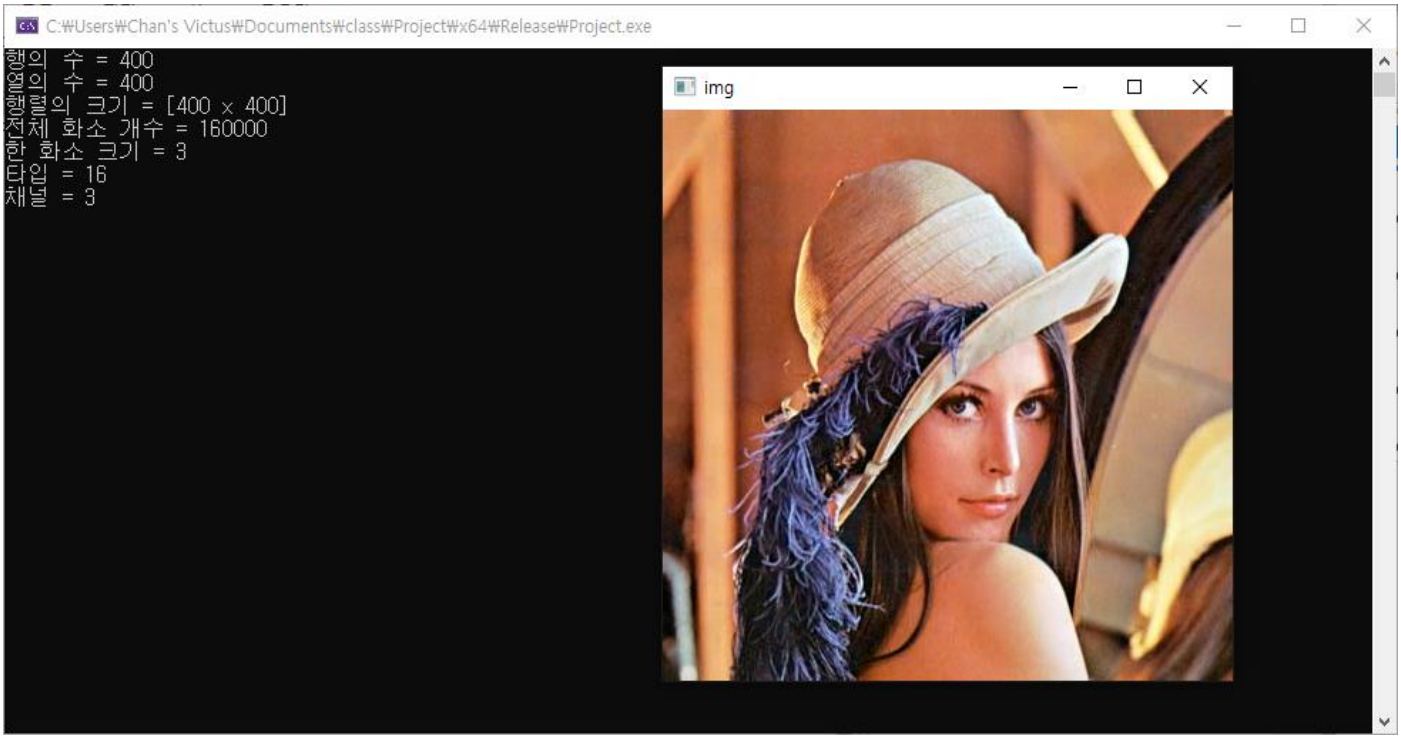
03-OpenCV의 기초(3) 26page

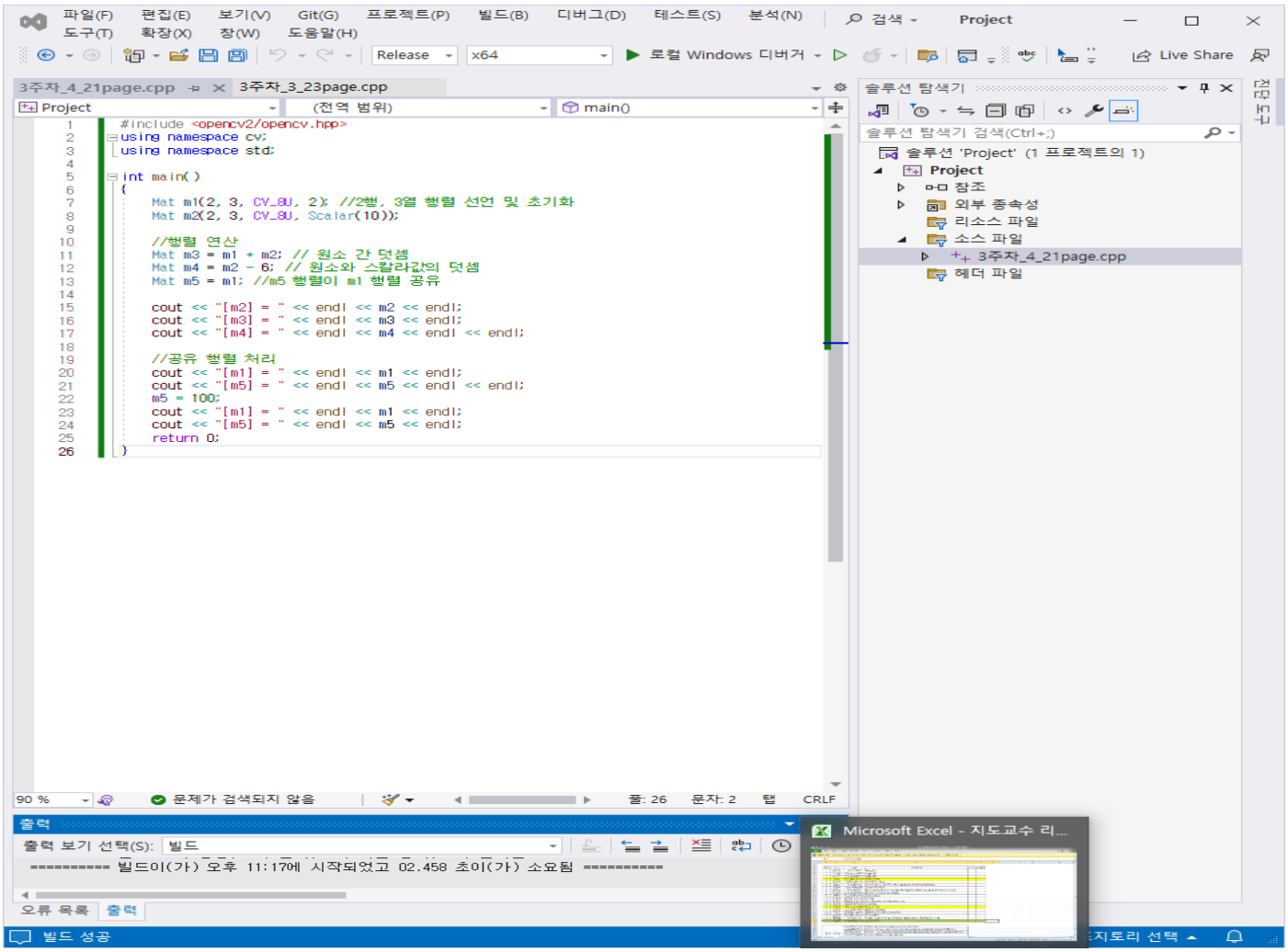
```cpp
#include <opencv2/opencv.hpp>
using namespace cv;
using namespace std;

int main()
{
    //VideoCapture cap(0); // 웹캠인 경우
    VideoCapture cap("../image/trailer.mp4");// 동영상 파일인 경우
    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;
    }
    return 0;
}
```

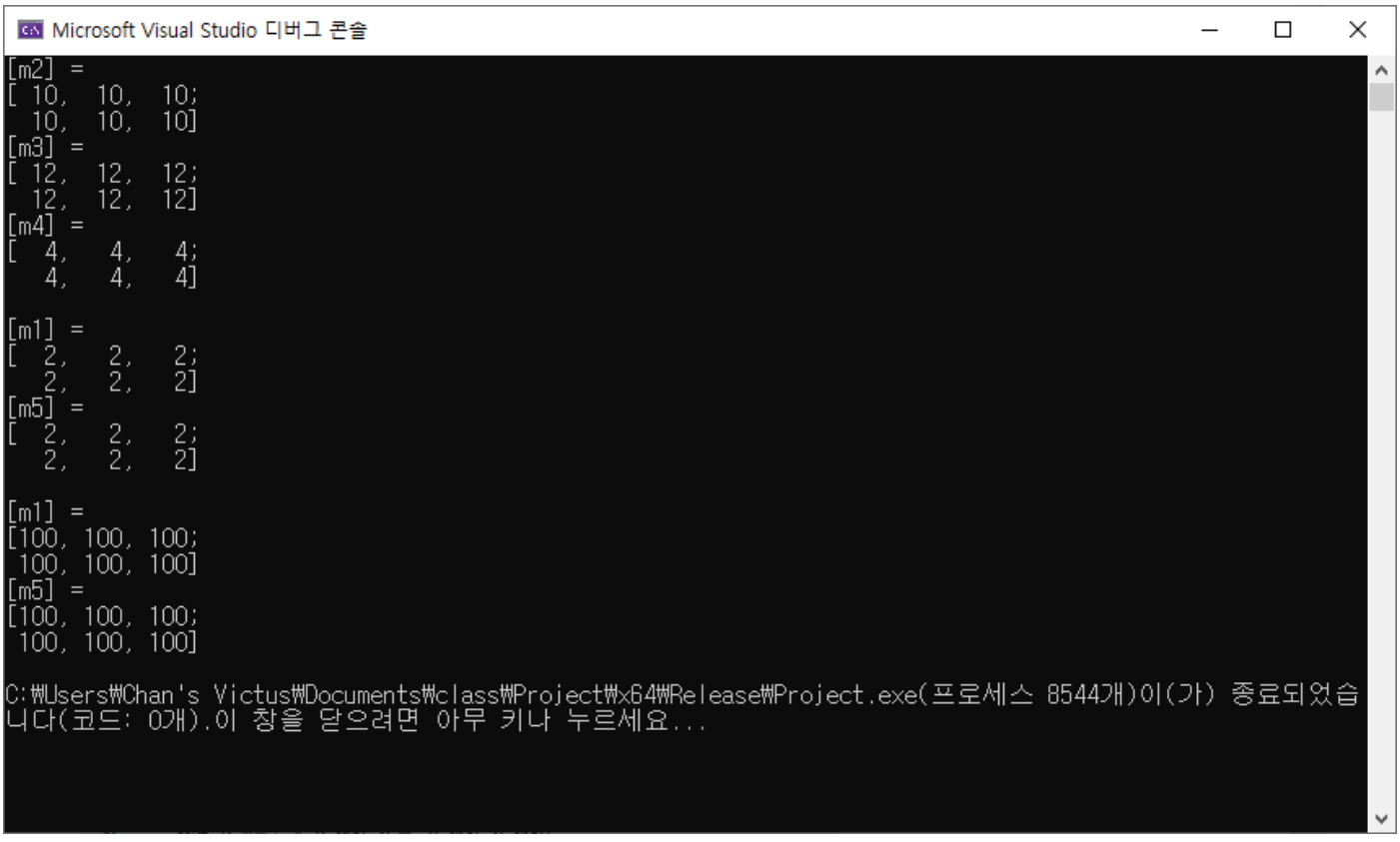



파일(F) 편집(E) 보기(V) Git(G) 프로젝트(P) 빌드(B) 디버그(D) 테스트(S) 분석(N)
도구(T) 확장(X) 창(W) 도움말(H)
검색
Project
Release
x64
로컬 Windows 디버거
Live Share
3주차_3_23page.cpp
3주차_4_6page.cpp
Project
(전역 범위)
main()
#include <opencv2/opencv.hpp>
using namespace cv;
using namespace std;
int main( )
{
float data[] = {
1.2f, 2.3f, 3.2f,
4.5f, 5.f, 6.5f,
};
//Mat 객체 선언 방법
Mat m1(2, 3, CV_8U);
Mat m2(2, 3, CV_8U, Scalar(300)); //uchar형 -> 255로 저장
Mat m3(2, 3, CV_16S, Scalar(300)); //uchar형 -> 300로 저장
Mat m4(2, 3, CV_32F, data); //배열 원소로 초기화
//Size_ 객체로 Mat 객체 선언방법
Size sz(2, 3);
Mat m5(Size(2, 3), CV_64F);
Mat m6(sz, CV_32F, data); //Size_ 객체로 초기화
cout << "[m1] =" << endl << m1 << endl;
cout << "[m2] =" << endl << m2 << endl;
cout << "[m3] =" << endl << m3 << endl;
cout << "[m4] =" << endl << m4 << endl;
cout << "[m5] =" << endl << m5 << endl;
cout << "[m6] =" << endl << m6 << endl;
return 0;
}
90 %
문제가 검색되지 않음
줄: 29 문자: 2 탭 CRLF
출력
출력 보기 선택(S): 빌드
========== 빌드이(가) 오후 11:06에 시작되었고 02.572 초이(가) 소요됨 ==========
오류 목록 출력
솔루션 탐색기
솔루션 탐색기 검색(Ctrl+;)
솔루션 'Project' (1 프로젝트의 1)
Project
참조
외부 종속성
리소스 파일
소스 파일
3주차_4_6page.cpp
헤더 파일
솔루션 탐색기 Git 변경 내용
빌드 성공
소스 제어에 추가
리포지토리 선택
Microsoft Visual Studio 디버그 콘솔
[m1] =
[0, 0, 0;
0, 0, 0]
[m2] =
[255, 255, 255;
255, 255, 255]
[m3] =
[300, 300, 300;
300, 300, 300]
[m4] =
[1.2, 2.3, 3.2;
4.5, 5, 6.5]
[m5] =
[0, 0;
0, 0;
0, 0]
[m6] =
[1.2, 2.3;
3.2, 4.5;
5, 6.5]
C:₩Users₩Chan's Victus₩Documents₩class₩Project₩x64₩Release₩Project.exe(프로세스 1044개)이(가) 종료되었습니다(코드: 0개).이 창을 닫으려면 아무 키나 누르세요...


3주차_4_17page.cpp
3주차_3_23page.cpp
Release
x64
로컬 Windows 디버거
Live Share
솔루션 탐색기
솔루션 'Project' (1 프로젝트의 1)
Project
참조
외부 종속성
리소스 파일
소스 파일
3주차_4_17page.cpp
헤더 파일
출력
빌드 성공


C:₩Users₩Chan's Victus₩Documents₩class₩Project₩x64₩Release₩Project.exe
행의 수 = 400
열의 수 = 400
행렬의 크기 = [400 x 400]
전체 화소 개수 = 160000
한 화소 크기 = 3
타입 = 16
채널 = 3
img

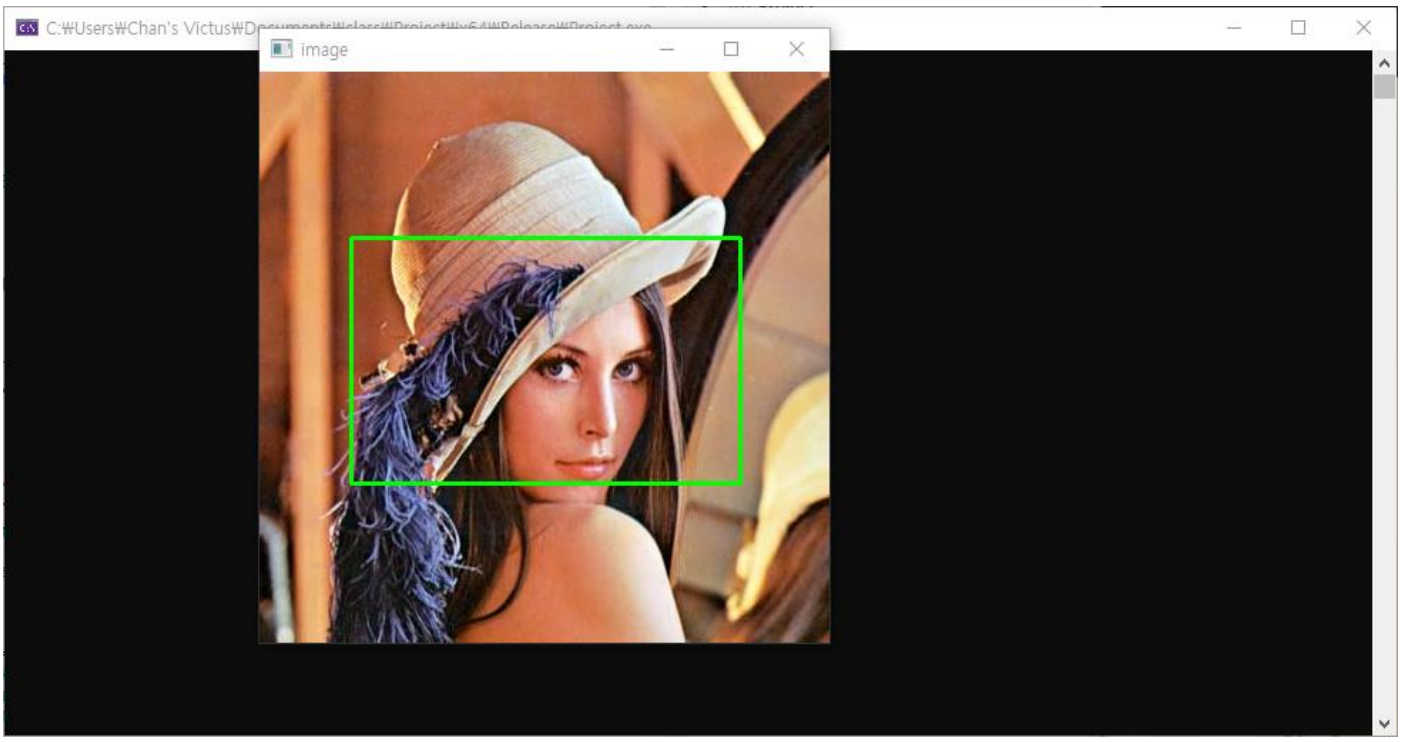

파일(F) 편집(E) 보기(V) Git(G) 프로젝트(P) 빌드(B) 디버그(D) 테스트(S) 분석(N)
도구(T) 확장(X) 창(W) 도움말(H)
검색
Project
Release
x64
로컬 Windows 디버거
Live Share
3주차_4_21page.cpp
3주차_3_23page.cpp
Project
(전역 범위)
main()
#include <opencv2/opencv.hpp>
using namespace cv;
using namespace std;
int main()
{
Mat m1(2, 3, CV_8U, 2); //2행, 3열 행렬 선언 및 초기화
Mat m2(2, 3, CV_8U, Scalar(10));
//행렬 연산
Mat m3 = m1 + m2; // 원소 간 덧셈
Mat m4 = m2 - 6; // 원소와 스칼라값의 덧셈
Mat m5 = m1; //m5 행렬이 m1 행렬 공유
cout << "[m2] = " << endl << m2 << endl;
cout << "[m3] = " << endl << m3 << endl;
cout << "[m4] = " << endl << m4 << endl << endl;
//공유 행렬 처리
cout << "[m1] = " << endl << m1 << endl;
cout << "[m5] = " << endl << m5 << endl << endl;
m5 = 100;
cout << "[m1] = " << endl << m1 << endl;
cout << "[m5] = " << endl << m5 << endl;
return 0;
}
솔루션 탐색기
솔루션 탐색기 검색(Ctrl+;)
솔루션 'Project' (1 프로젝트의 1)
Project
참조
외부 종속성
리소스 파일
소스 파일
3주차_4_21page.cpp
헤더 파일
90 %
문제가 검색되지 않음
줄: 26 문자: 2 탭 CRLF
출력
출력 보기 선택(S): 빌드
========== 빌드이(가) 오후 11:17에 시작되었고 02.458 초이(가) 소요됨 ==========
오류 목록 출력
빌드 성공
Microsoft Excel - 지도교수 리...

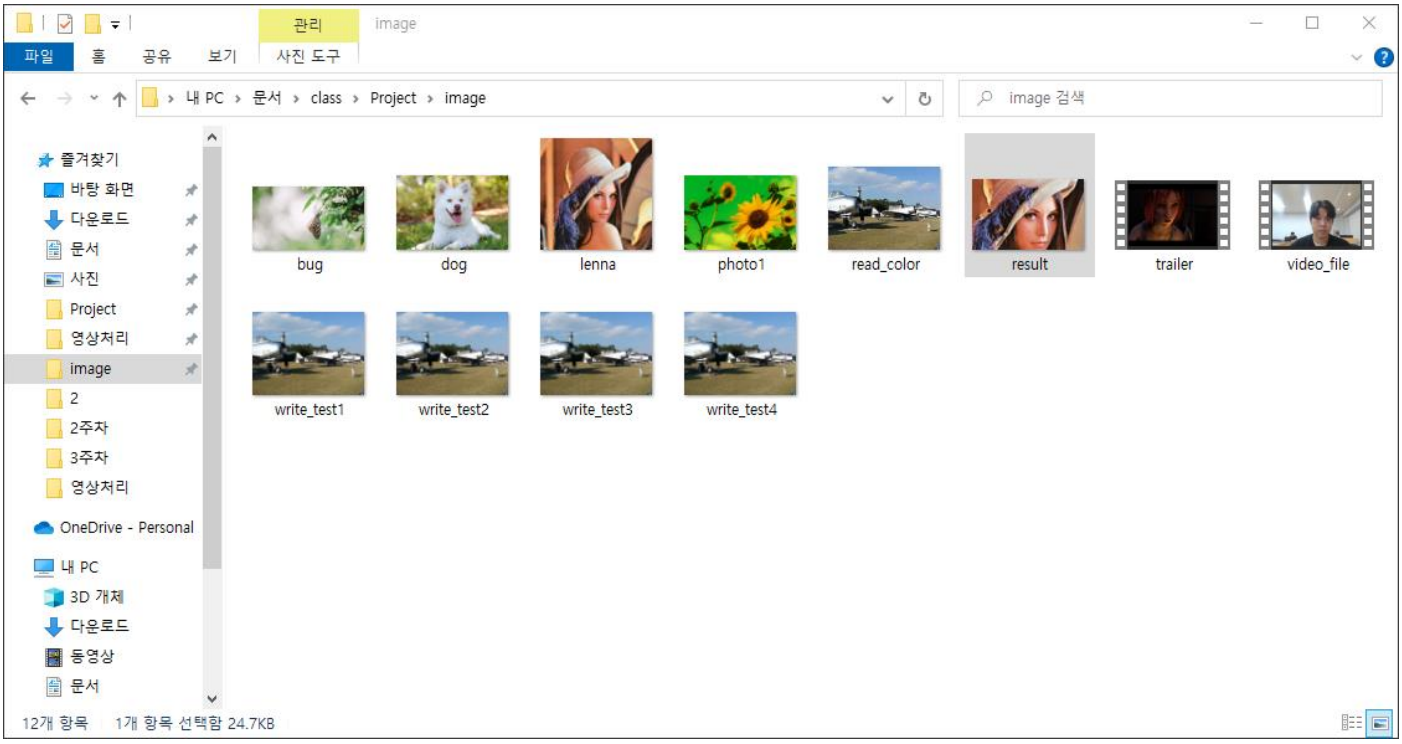

Microsoft Visual Studio 디버그 콘솔
[m2] =
[10, 10, 10;
10, 10, 10]
[m3] =
[12, 12, 12;
12, 12, 12]
[m4] =
[4, 4, 4;
4, 4, 4]
[m1] =
[2, 2, 2;
2, 2, 2]
[m5] =
[2, 2, 2;
2, 2, 2]
[m1] =
[100, 100, 100;
100, 100, 100]
[m5] =
[100, 100, 100;
100, 100, 100]
C:₩Users₩Chan's Victus₩Documents₩class₩Project₩x64₩Release₩Project.exe(프로세스 8544개)이(가) 종료되었습니다(코드: 0개).이 창을 닫으려면 아무 키나 누르세요...

03-OpenCV의 기초(4) 33page

```cpp
#include "opencv2/opencv.hpp"
#include <iostream>
using namespace cv;
using namespace std;

Mat img, roi;
int mx1, my1, mx2, my2; // 마우스로 지정한 사각형의 좌표
bool cropping = false; // 사각형 선택 중임을 나타내는 플래그 변수

//마우스 이벤트가 발생하면 호출되는 콜백 함수이다.
void onMouse (int event, int x, int y, int flags, void* param)
{
    if (event == EVENT_LBUTTONDOWN) {// 마우스의 왼쪽 버튼을 누르면
            mx1 = x; // 사각형의 좌측 상단 좌표 저장
            my1 = y;
            cropping = true;
        }
    else if (event == EVENT_LBUTTONUP) {// 마우스의 왼쪽 버튼에서 손을 떼면
            mx2 = x; // 사각형의 우측 하단 좌표 저장
            my2 = y;
            cropping = false;
            rectangle(img, Rect(mx1, my1, mx2-mx1, my2-my1), Scalar(0, 255, 0), 2);
            imshow("image", img);
        }
    }

int main() {
    img = imread("../image/lenna.jpg");
    imshow("image", img);
    Mat clone = img.clone(); // 복사본을 만들어둔다

    setMouseCallback("image", onMouse);

    while (1) {
        int key = waitKey(100);
        if (key == 'q') break; // 사용자가 'q' 를 누르면 종료
        else if (key == 'c') {   // 사용자가 'c' 를 누르면 관심영역을 파일로 저장
                roi = clone(Rect(mx1, my1, mx2-mx1, my2-my1));
                imwrite("../image/result.jpg", roi);
            }
    }
    return 0;
}
```

₩




관리
image
파일
홈
공유
보기
사진 도구
내 PC › 문서 › class › Project › image
image 검색
즐겨찾기
바탕 화면
다운로드
문서
사진
Project
영상처리
image
2
2주차
3주차
영상처리
OneDrive - Personal
내 PC
3D 개체
다운로드
동영상
문서
bug
dog
lenna
photo1
read_color
result
trailer
video_file
write_test1
write_test2
write_test3
write_test4
12개 항목
1개 항목 선택함 24.7KB