

## 2장 OPENCV 설치와 개요

1

### OpenCV란?

- 컴퓨터 비전 응용 프로그램을 개발할 수 있는 강력한 라이브러리이다.
- OpenCV는 BSD 라이선스 하에 배포되므로 학술적 및 상업적 용도로 무료이다.
- OpenCV는 계산 효율성과 실시간 응용 프로그램에 중점을 두고 설계되었다.

OpenCV로 배우는 디지털 영상 처리

© 인피니티북스 2019



2

# OpenCV의 구성

## 영상 처리

필터링

에지검출

변환

## 특징 분석

물체 감지

트래킹

얼굴 인식

## 기타


기계 학습

영상 피라미드

스테레오 비전

OpenCV로 배우는 디지털 영상 처리

© 인피니티박스 2019




3

# OpenCV로 할 수 있는 작업

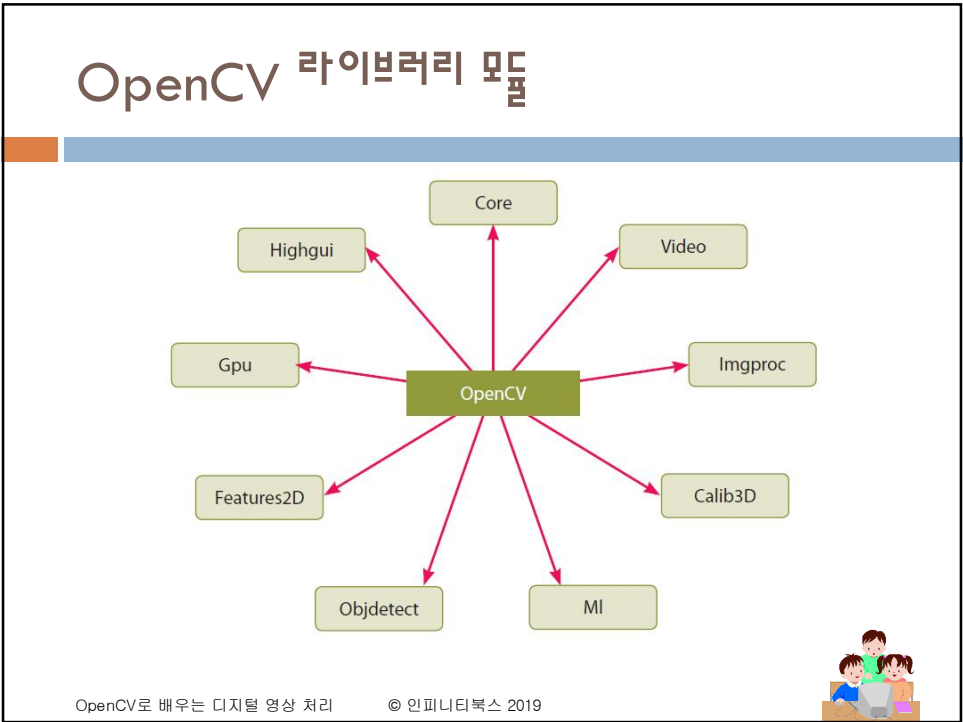
- 영상 파일의 읽기 및 쓰기
- 비디오 캡처 및 저장
- 영상 처리(필터, 변환)
- 영상이나 비디오에서 얼굴, 눈, 자동차와 같은 특정 물체를 감지
- 비디오를 분석하여 움직임 추정하고, 배경을 없애고, 특정 물체를 추적할 수 있다.
- 기계 학습 알고리즘을 사용하여 물체를 인식할 수 있다.

OpenCV로 배우는 디지털 영상 처리

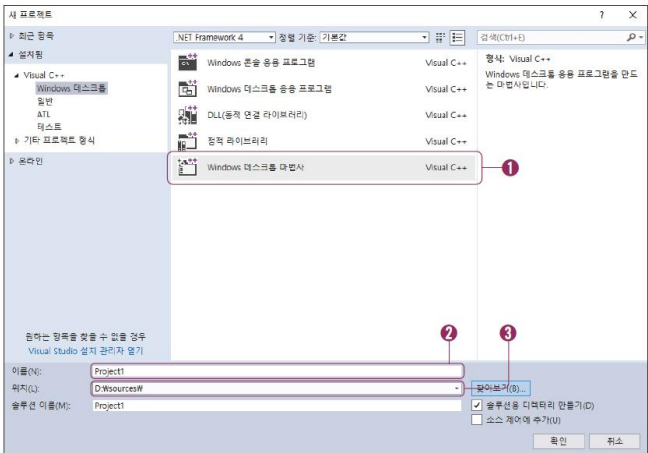
© 인피니티박스 2019



4



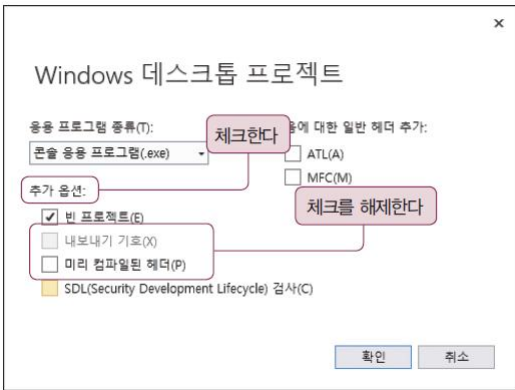
# OpenCV 사용하기



OpenCV로 배우는 디지털 영상 처리 © 인피니티북스 2019

7

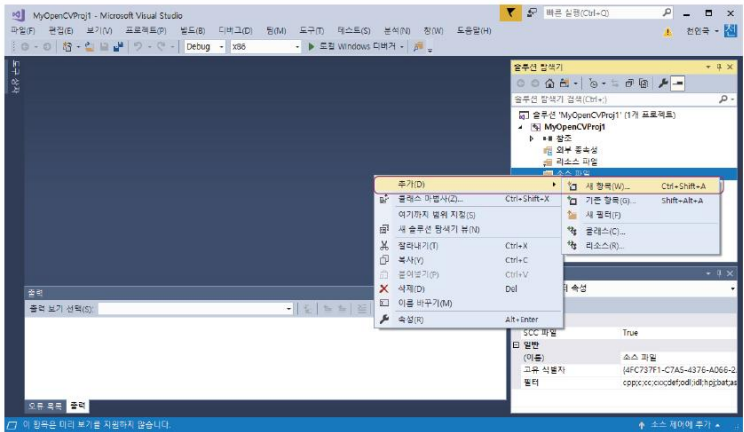
# OpenCV 사용하기



OpenCV로 배우는 디지털 영상 처리 © 인피니티북스 2019

8

# OpenCV 사용하기



OpenCV로 배우는 디지털 영상 처리      © 인피니티북스 2019



# OpenCV 사용하기

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

using namespace cv;
using namespace std;

int main()
{
    Mat image;
    image = imread("d:/lenna.jpg", IMREAD_COLOR);
    if (image.empty()) { cout << "영상을 읽을 수 없음" << endl; }

    imshow("출력 영상", image);
    waitKey(0);
    return 0;
}
```

OpenCV로 배우는 디지털 영상 처리      © 인피니티북스 2019



## OpenCV 사용하기

□ 많은 오류 발생!

1. OpenCV 헤더 파일의 위치를 컴파일러에게 알려주어야 한다.
2. OpenCV 라이브러리가 설치된 위치를 링커에게 알려 주어야 한다.

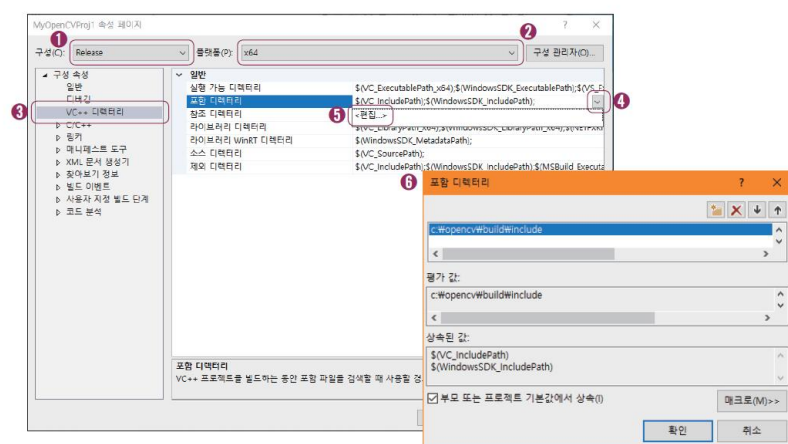
OpenCV로 배우는 디지털 영상 처리

© 인피니티북스 2019



11

## OpenCV 사용하기



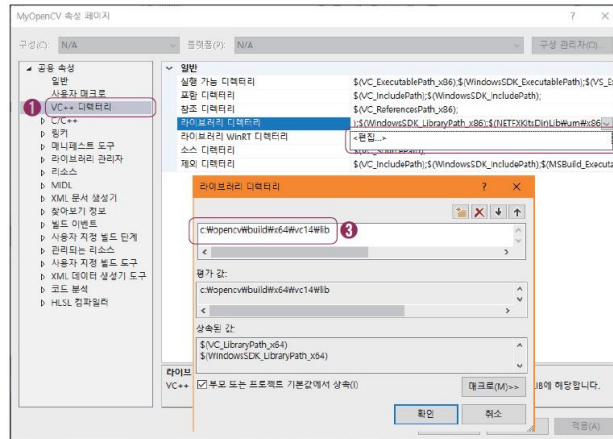
OpenCV로 배우는 디지털 영상 처리

© 인피니티북스 2019



12

## OpenCV 사용하기

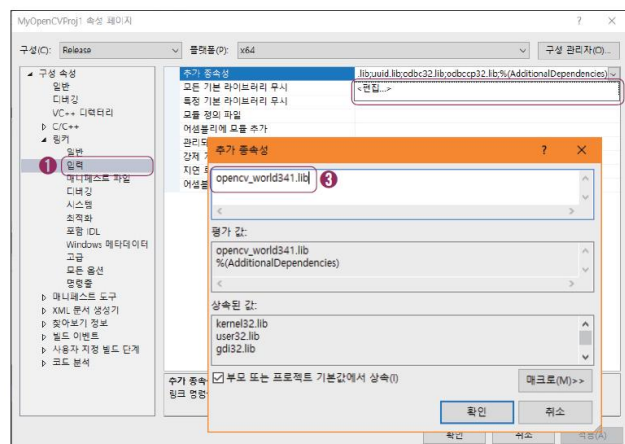


OpenCV로 배우는 디지털 영상 처리 © 인피니티북스 2019



13

## OpenCV 사용하기



OpenCV로 배우는 디지털 영상 처리 © 인피니티북스 2019



14

## OpenCV 사용하기

파일명	확장자	크기	날짜	속성
[.]		<폴더>	2018-04-22 15:28	—
opencv_annotation	exe	53,760	2018-02-23 22:46	-a-
opencv_createsamples	exe	57,344	2018-02-23 22:46	-a-
opencv_ffmpeg341_64	dll	18,053,632	2018-02-23 22:41	-a-
opencv_interactive-calibration	exe	139,776	2018-02-23 22:46	-a-
opencv_traincascade	exe	329,728	2018-02-23 22:46	-a-
opencv_version	exe	39,424	2018-02-23 22:46	-a-
opencv_visualization	exe	61,440	2018-02-23 22:46	-a-
opencv_world341	dll	65,811,968	2018-02-23 22:46	-a-
opencv_world341d	dll	104,591,872	2018-02-23 22:55	-a-

파일명	확장자	크기	날짜	속성
[.]		<폴더>	2018-07-29 16:37	—
[x64]		<폴더>	2018-07-29 16:36	—
test	cpp	303	2018-07-29 16:37	-a-
opencv_world341	dll	65,811,968	2018-02-23 22:46	-a-
Project1.vcxproj	filters	961	2018-07-29 16:36	-a-
Project1	vcxproj	7,516	2018-07-29 16:36	-a-

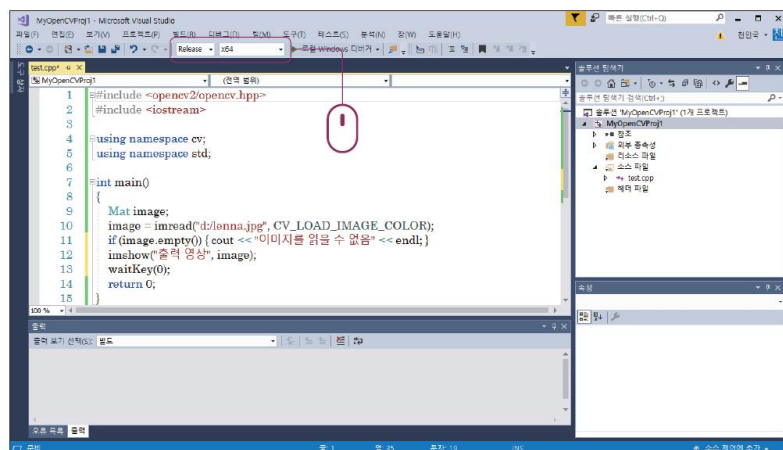
OpenCV로 배우는 디지털 영상 처리

© 인피니티북스 2019



15

## OpenCV 사용하기



OpenCV로 배우는 디지털 영상 처리

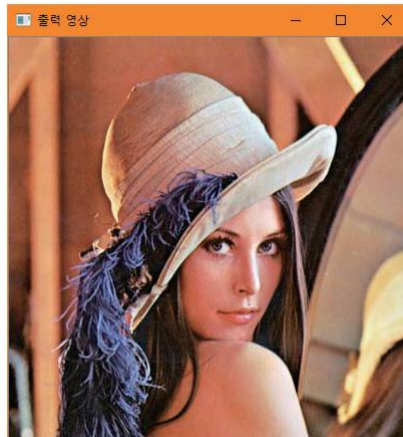
© 인피니티북스 2019



16



## OpenCV 사용하기



OpenCV로 배우는 디지털 영상 처리

© 인피니티북스 2019



17

## OpenCV의 속성 시트

- 속성을 XML 파일로 저장하여서 차후 프로젝트에서 사용할 수 있는 방법

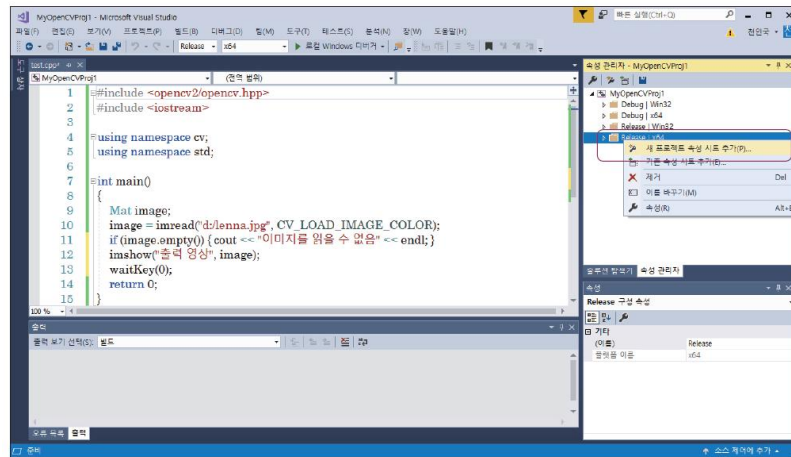
OpenCV로 배우는 디지털 영상 처리

© 인피니티북스 2019



18

## OpenCV의 속성 시트



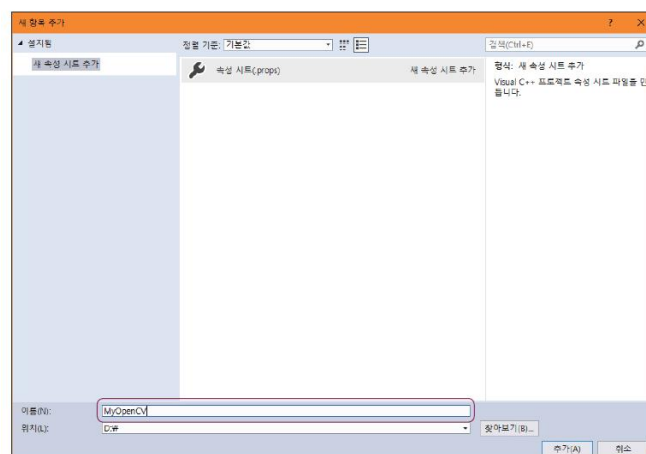
OpenCV로 배우는 디지털 영상 처리

© 인피니티북스 2019



19

## OpenCV의 속성 시트



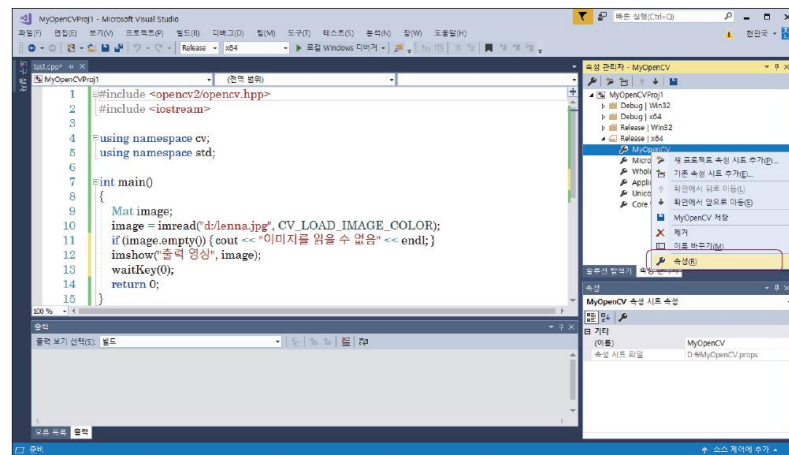
OpenCV로 배우는 디지털 영상 처리

© 인피니티북스 2019



20

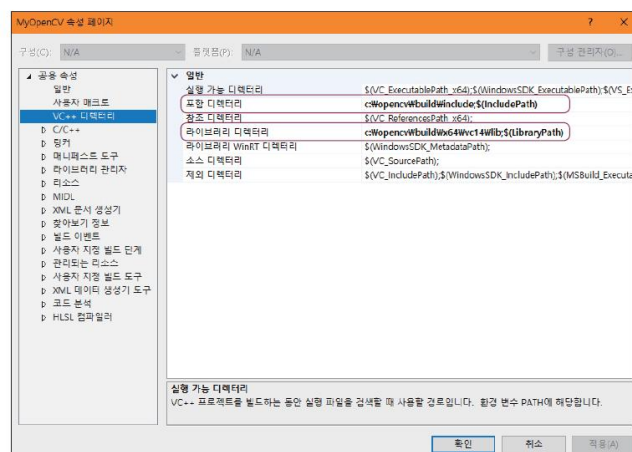
## OpenCV의 속성 시트



OpenCV로 배우는 디지털 영상 처리 © 인피니티북스 2019

21

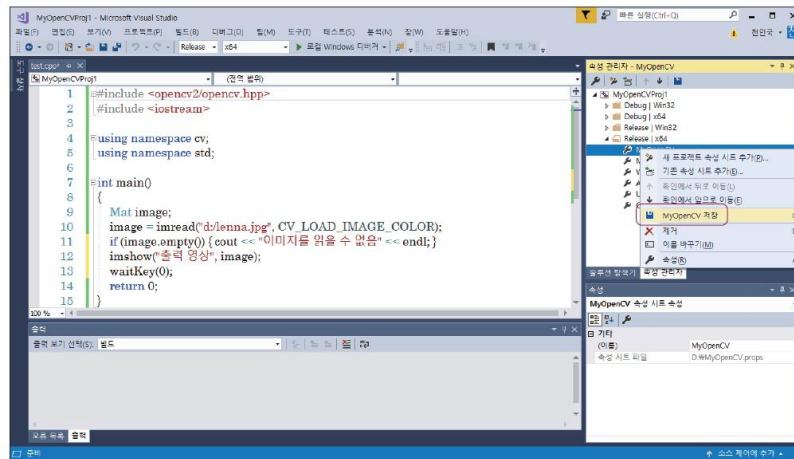
## OpenCV의 속성 시트



OpenCV로 배우는 디지털 영상 처리 © 인피니티북스 2019

22

## OpenCV의 속성 시트



OpenCV로 배우는 디지털 영상 처리 © 인피니티북스 2019



23

## OpenCV의 속성 시트



OpenCV로 배우는 디지털 영상 처리 © 인피니티북스 2019



24

## 소스 분석

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

using namespace cv;
using namespace std;

int main()
{
    Mat image;
    image = imread("d:/lenna.jpg", IMREAD_COLOR);

    if( image.empty() ) { cout << "영상을 읽을 수 없음" << endl; }
    imshow("출력 영상", image);

    waitKey(0);
    return 0;
}
```

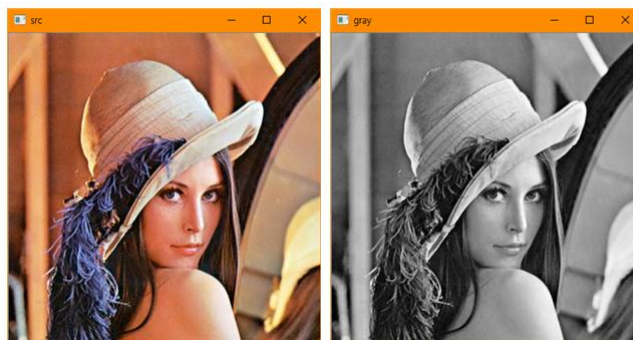
OpenCV로 배우는 디지털 영상 처리

© 인피니티박스 2019



25

## 간단한 영상 처리 경험해보기



OpenCV로 배우는 디지털 영상 처리

© 인피니티박스 2019



26

## 소스 분석

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

int main()
{
    Mat src = imread("d:/lenna.jpg", IMREAD_COLOR);
    if (src.empty()) { cout << "영상을 읽을 수 없음" << endl; }
    imshow("src", src);

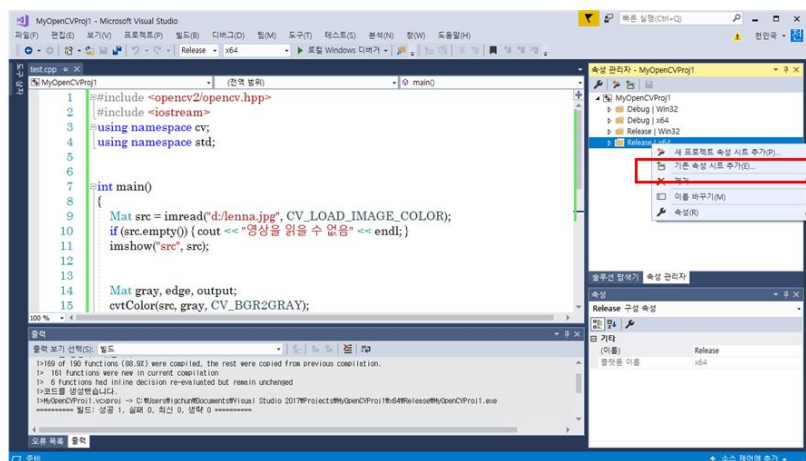
    Mat gray, edge, output;
    cvtColor(src, gray, CV_BGR2GRAY);

    imshow("gray", gray);
    imwrite("d:/gray.jpg", gray);
    waitKey(0);
    return 0;
}
```

OpenCV로 배우는 디지털 영상 처리 © 인피니티북스 2019

27

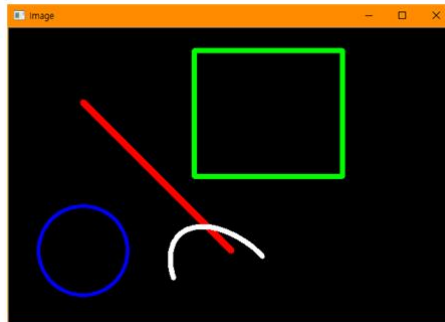
## 소스 실행



OpenCV로 배우는 디지털 영상 처리 © 인피니티북스 2019

28

## OpenCV를 이용하여 도형 그리기



OpenCV로 배우는 디지털 영상 처리

© 인피니티북스 2019



29

## 소스 분석

```
#include <opencv2/core.hpp>
#include <opencv2/imgproc.hpp>
#include <opencv2/highgui.hpp>
using namespace cv;
using namespace std;

int main()
{
    // 검정색으로 초기화된 600×400 크기의 영상 생성
    Mat image = Mat(400, 600, CV_8UC3, Scalar(0, 0, 0));

    line(image, Point(100, 100), Point(300, 300), Scalar(0, 0, 255), 7);
    rectangle(image, Point(250, 30), Point(450, 200), Scalar(0, 255, 0), 5);
    circle(image, Point(100, 300), 60, Scalar(255, 0, 0), 3);
    ellipse(image, Point(300, 350), Point(100, 60), 45, 130, 270, Scalar(255, 255, 255), 5);

    imshow("Image", image);
    waitKey(0);
    return(0);
}
```

OpenCV로 배우는 디지털 영상 처리

© 인피니티북스 2019



30

## 키보드 및 마우스 이벤트 처리하기



OpenCV로 배우는 디지털 영상 처리

© 인피니티북스 2019



31

## 소스 분석

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

// 마우스 이벤트가 발생하면 호출되는 콜백 함수이다.
void onMouse(int event, int x, int y, int flags, void* param)
{
    if (event == EVENT_LBUTTONDOWN){
        Mat& img = *(Mat*)(param);
        circle(img, Point(x, y), 200, Scalar(0, 255, 0), 10);
        putText(img, "I found a dog!", Point(x, y + 200),
        FONT_HERSHEY_PLAIN, 2.0, 255, 2);
        imshow("src", img); // 영상이 변경되면 다시 표시한다.
    }
    else if (event == EVENT_RBUTTONDOWN){}
    else if (event == EVENT_MBUTTONDOWN){}
    else if (event == EVENT_MOUSEMOVE){}
}
}
```

OpenCV로 배우는 디지털 영상 처리

© 인피니티북스 2019



32



## 소스 분석

```
int main()
{
    Mat src = imread("d:/dog.jpg", IMREAD_COLOR);
    if (src.empty()) { cout << "영상을 읽을 수 없음" << endl; }
    imshow("src", src);

    setMouseCallback("src", onMouse, &src);
    waitKey(0);
    return 0;
}
```

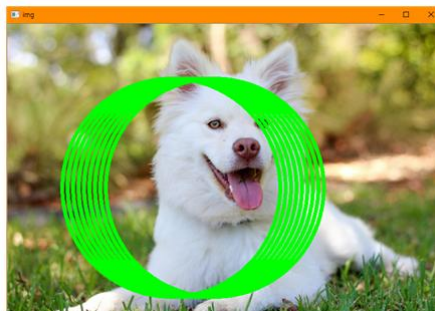
OpenCV로 배우는 디지털 영상 처리

© 인피니티북스 2019



33

## 키보드 이벤트 처리하기



OpenCV로 배우는 디지털 영상 처리

© 인피니티북스 2019



34

## 소스 분석

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

using namespace std;
using namespace cv;

int main()
{
    Mat img;
    img = imread("d:/dog.jpg", IMREAD_COLOR);
    if (img.empty()) { cout << "영상을 읽을 수 없음" << endl; }

    imshow("img", img);
    int x = 300;
    int y = 300;
```

OpenCV로 배우는 디지털 영상 처리 © 인피니티북스 2019

35

## 소스 분석

```
while (1) {

    int key = waitKey(100);
    if (key == 'q') break;
    else if (key == 'a')
        x -= 10;
    else if (key == 'w')
        y -= 10;
    else if (key == 'd')
        x += 10;
    else if (key == 's')
        y += 10;

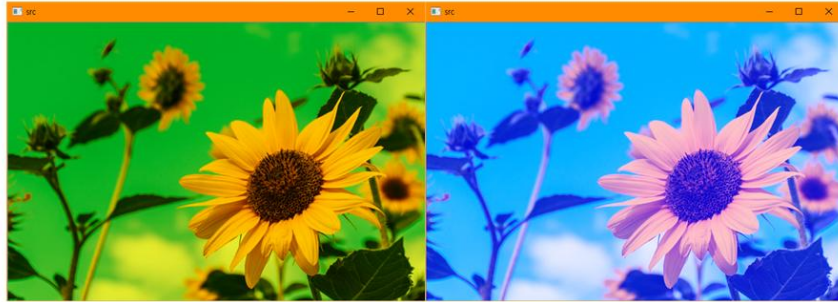
    circle(img, Point(x, y), 200, Scalar(0, 255, 0), 5);
    imshow("img", img);

}
return 0;
}
```

OpenCV로 배우는 디지털 영상 처리 © 인피니티북스 2019

36

## 키보드로 영상 제어하기



OpenCV로 배우는 디지털 영상 처리

© 인피니티북스 2019



37

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

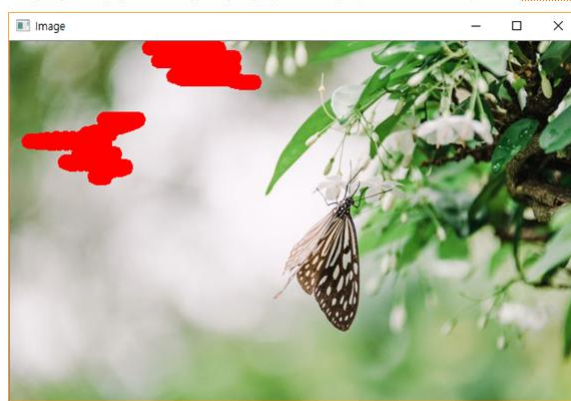
int main()
{
    Mat src = imread("d:/photo1.jpg", IMREAD_COLOR);
    if (src.empty()) { cout << "영상을 읽을 수 없음" << endl; }

    imshow("src", src);

    while (1) {
        int key = waitKeyEx(); // 사용자로부터 키를 기다린다.
        cout << key << " ";
        if (key == 'q') break; // 사용자가 'q'를 누르면 종료한다.
        else if (key == 2424832) {
            src -= 50; // 영상이 어두워진다.
        }
        else if (key == 2555904) {
            src += 50; // 영상이 밝아진다.
        }
        imshow("src", src); // 영상이 변경되었으므로 다시 표시한다.
    }
    return 0;
}
```

38

## 페인트 브러시 프로그램 작성



OpenCV로 배우는 디지털 영상 처리

© 인피니티북스 2019



39

## 소스 분석

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

Mat img;
int drawing = false;

void drawCircle(int event, int x, int y, int, void* param) {
    if (event == CV_EVENT_LBUTTONDOWN)
        drawing = true;
    else if (event == CV_EVENT_MOUSEMOVE) {
        if (drawing == true)
            circle(img, Point(x, y), 3, Scalar(0, 0, 255), 10);
    }
    else if (event == CV_EVENT_LBUTTONUP)
        drawing = false;
    imshow("Image", img);
}
```

40

# 소스 분석

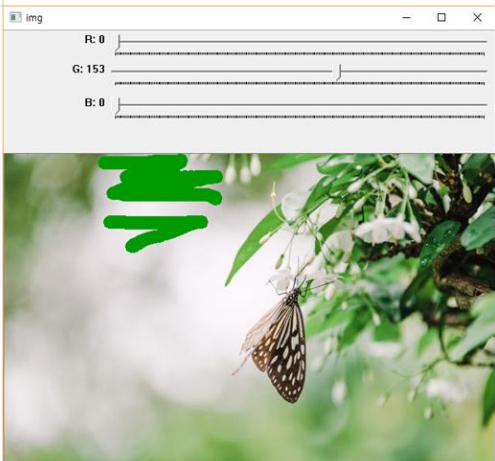
```
int main()
{
    img = imread("d:/bug.jpg");
    if (img.empty()) { cout << "영상을 읽을 수 없음" << endl; return
-1; }

    imshow("Image", img);

    setMouseCallback("Image", drawCircle);
    waitKey(0);
    imwrite("d:/bug1.jpg", img);
    return 0;
}
```

41

# 트랙바의 사용



42

### ▲ ▲ H A J

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

Mat img;
int red, green, blue;
int drawing = false;

void on_trackbar(int, void*) { }

void drawCircle(int event, int x, int y, int, void* param) {
    if (event == CV_EVENT_LBUTTONDOWN)
        drawing = true;
    else if (event == CV_EVENT_MOUSEMOVE) {
        if (drawing == true)
            circle(img, Point(x, y), 3, Scalar(blue, green, red), 10);
    }
    else if (event == CV_EVENT_LBUTTONUP)
        drawing = false;
    imshow("img", img);
}
```

43

### 소스 분석

```
int main()
{
    img = imread("d:/bug.jpg");
    if (img.empty()) { cout << "영상을 읽을 수 없음" << endl; return
-1; }

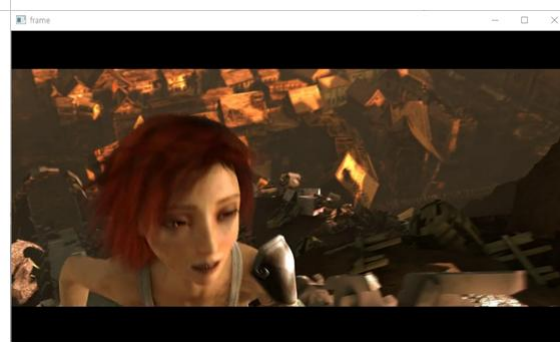
    namedWindow("img", 1);
    imshow("img", img);
    setMouseCallback("img", drawCircle);
    createTrackbar("R", "img", &red, 255, on_trackbar);
    createTrackbar("G", "img", &green, 255, on_trackbar);

    createTrackbar("B", "img", &blue, 255, on_trackbar);
    waitKey(0);
    return 0;
}
```



44

## 비디오 처리



OpenCV로 배우는 디지털 영상 처리

© 인피니티북스 2019



45

## 소스 분석

```
#include "opencv2/opencv.hpp"
using namespace cv;

int main()
{
    // VideoCapture cap(0);          // 웹캠인 경우
    VideoCapture cap("d:/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 추출한
        다.

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

OpenCV로 배우는 디지털 영상 처리

© 인피니티북스 2019



46

Q & A

