

# **Image Processing**

- Image, Video I/O using OpenCV**

---

# Contents

---

- ☐ Introduction
- ☐ OpenCV (Open Source Computer Vision)
- ☐ Install & Setting OpenCV
- ☐ Image I/O using OpenCV
- ☐ Video I/O using OpenCV

# Introduction

---

- 수업 시간 : 목요일 0, 1, 2교시
  
- 수업 내용
  - 영상처리 관련 실습
  - MFC를 이용한 Viewer 및 영상 분석 도구 제작

# Introduction

---

## □ 수업 내용

- 1. Image, Video In/Out With OpenCV
- 2. Color Conversion
- 3. Interpolation & Histogram
- 4. Spatial Filtering
- 5. Edge, Canny Edge
- 6. Color Image Processing, Morphological Filtering
- 7. Feature Detecting, Matching
- 8. Image Transform & RANSAC
- 9. Segmentation
- 10. Deep Learning

# Introduction

## □ 수업 일정

날짜	수업 내용	비고
9 / 4	1	
9 / 11	2	
9 / 18	3	
9 / 25	4	
9 / 28	제안서 발표	
10 / 16	5	
10 / 23	중간고사	
10 / 30	6	
11 / 6	중간 발표	
11 / 13	7	
11 / 20	8	
11 / 27	9	
11 / 4	10	
12 / 11	기말고사	
12 / 18	최종발표	

# **OpenCV (Open Source Computer Vision)**

---

# OpenCV (Open Source Computer Vision)

---

## □ OpenCV (Open Source Computer Vision Library)

- Intel사에 의해 1999년 처음 발표된 영상처리 및 컴퓨터 비전 라이브러리
- Open Source Project로 개발
- 현재는 OpenCV Foundation에서 프로젝트 관리가 이루어지고 있음
- 4.4.0 Version Release

### ■ 주요 기능

- 영상의 파일 및 화면 입/출력, 기본 연산과 행렬 단위 연산
- 영상의 광학적 변환, 다양한 공간적 필터 처리
- 영상의 기하학적 변환
- 영상 색 공간 변환과 영상 분할
- 히스토그램 분석 및 처리
- 기계 학습에 기반을 둔 물체 인식

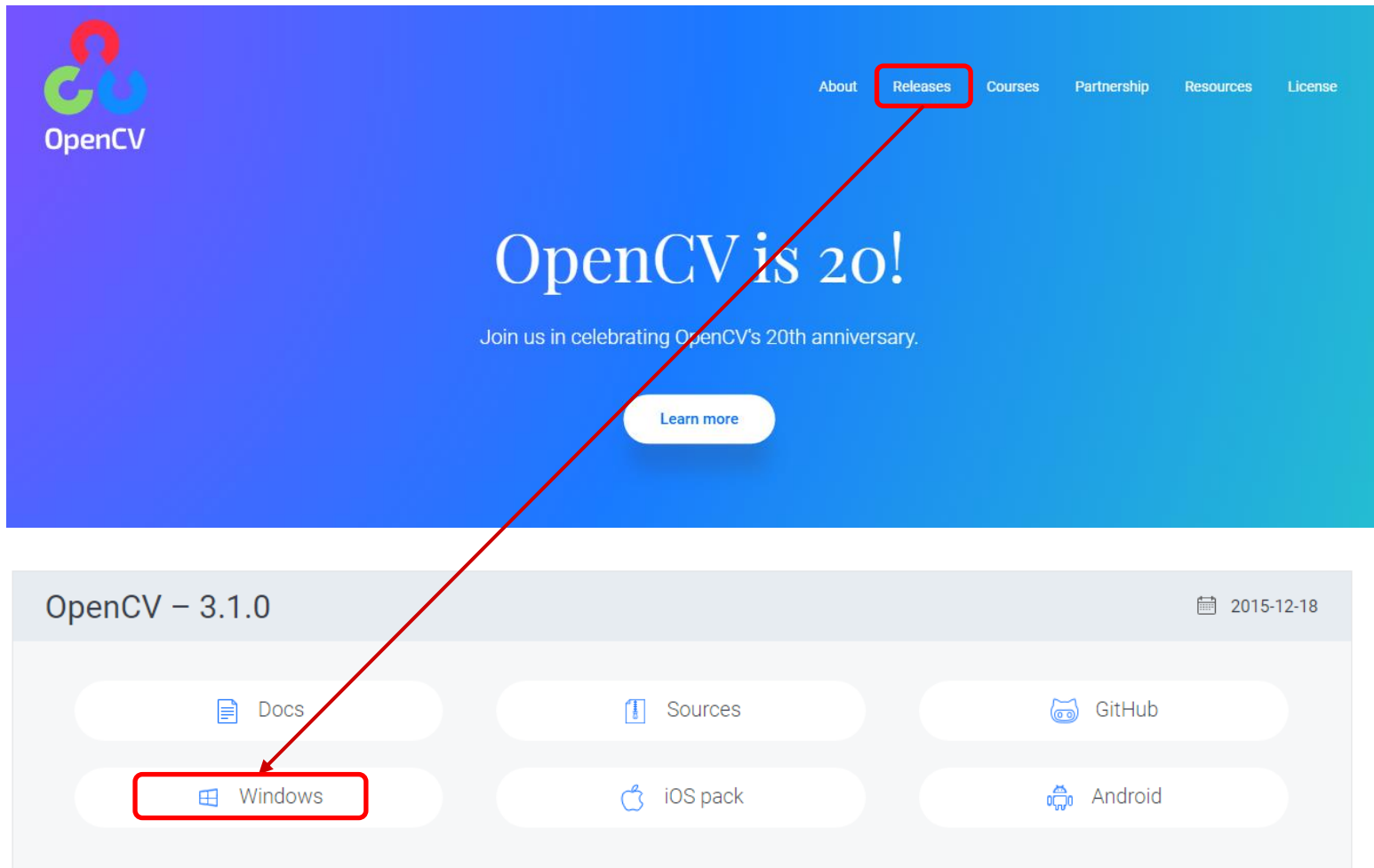
# **Install & Setting**

---



# Install & Setting

- Homepage - <http://opencv.org/>



# Install & Setting

## □ 3.1 Version 다운로드

**SOURCEFORGE** Search Browse Enterprise Blog Articles Deals Help Create Log In or Join

SOLUTION CENTERS Resources Newsletters Cloud Storage Providers Business VoIP Providers Internet Speed Test Call Center Providers

**에듀윌** 전기기사 합격완성풀서비스 시험 직전! 핵심150제 제공!

EDIYA COFFEE 30% 할인

**OpenCV**  
Download of opencvlibrary will start in 0 seconds...

Problems with the download? Please use this [direct link](#), or try another [mirror](#).

Mirror provided by: **JAIST 北陸**  
北陸先端科学技術大学院大学

Scanned by: **Bitdefender**

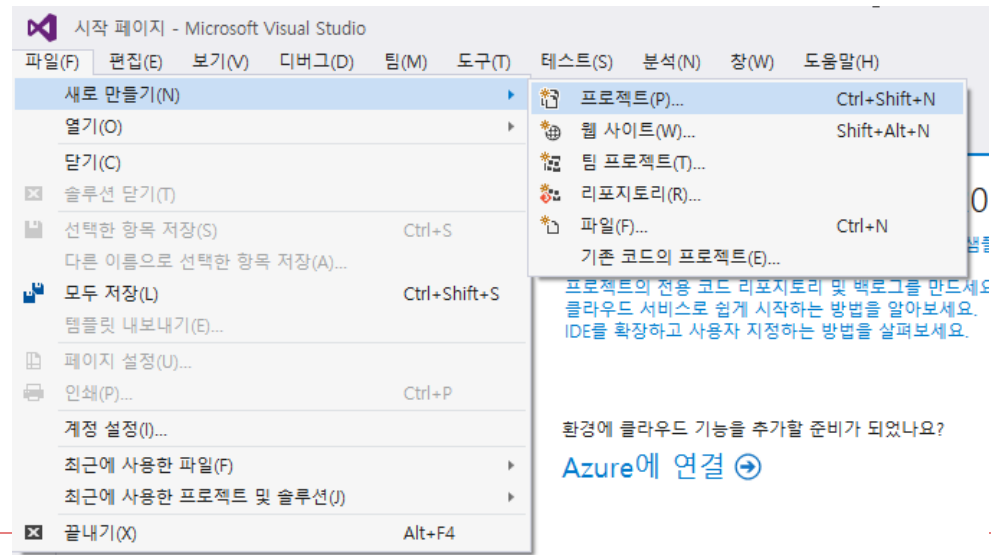


# Install & Setting

## □ 압축 해제 (C:\W)

내 PC > 로컬 디스크 (C:) > opencv				
이름	수정한 날짜	유형	크기	
build	2016-04-01 오전...	파일 폴더		
sources	2016-04-01 오전...	파일 폴더		
LICENSE.txt	2015-06-04 오전...	텍스트 문서	3KB	
README.md.txt	2015-06-04 오전...	텍스트 문서	1KB	

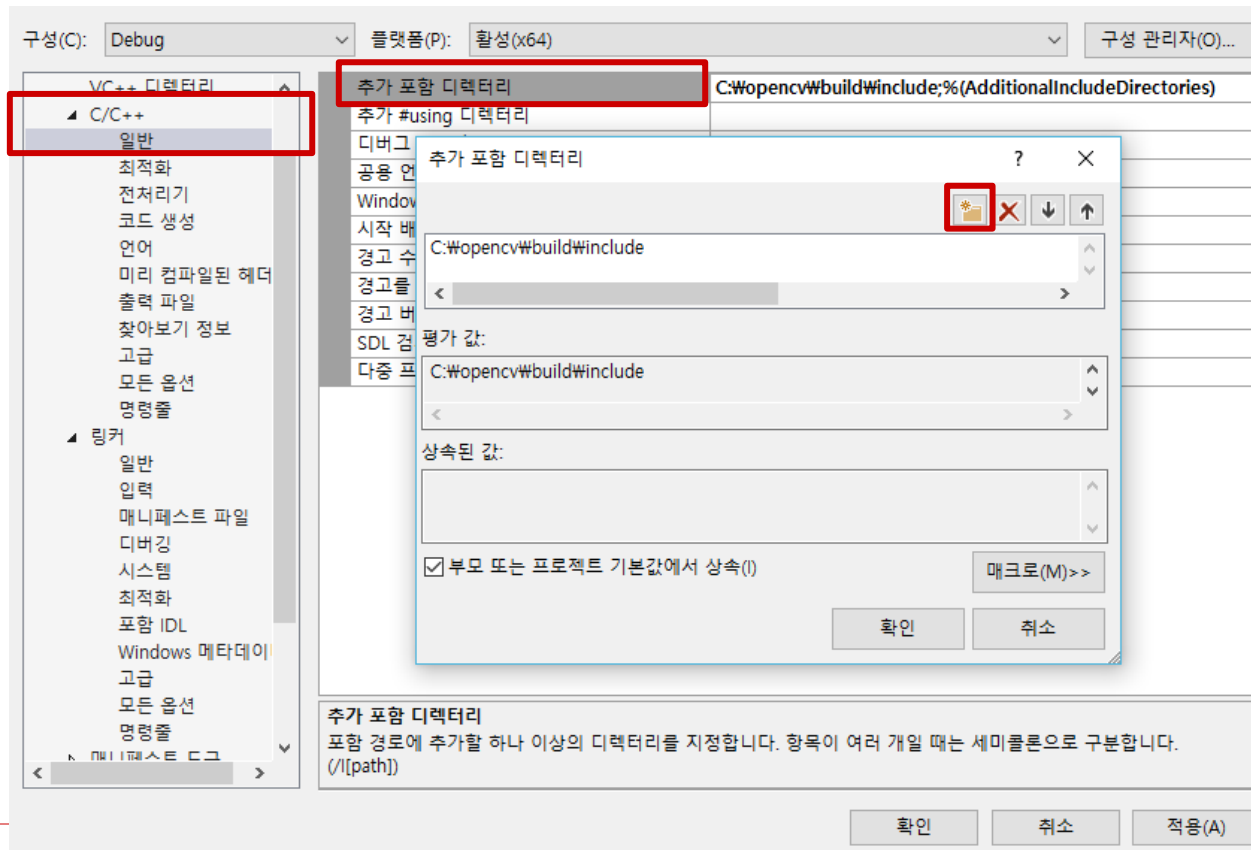
## □ 프로젝트 생성



# Install & Setting

## □ Setting

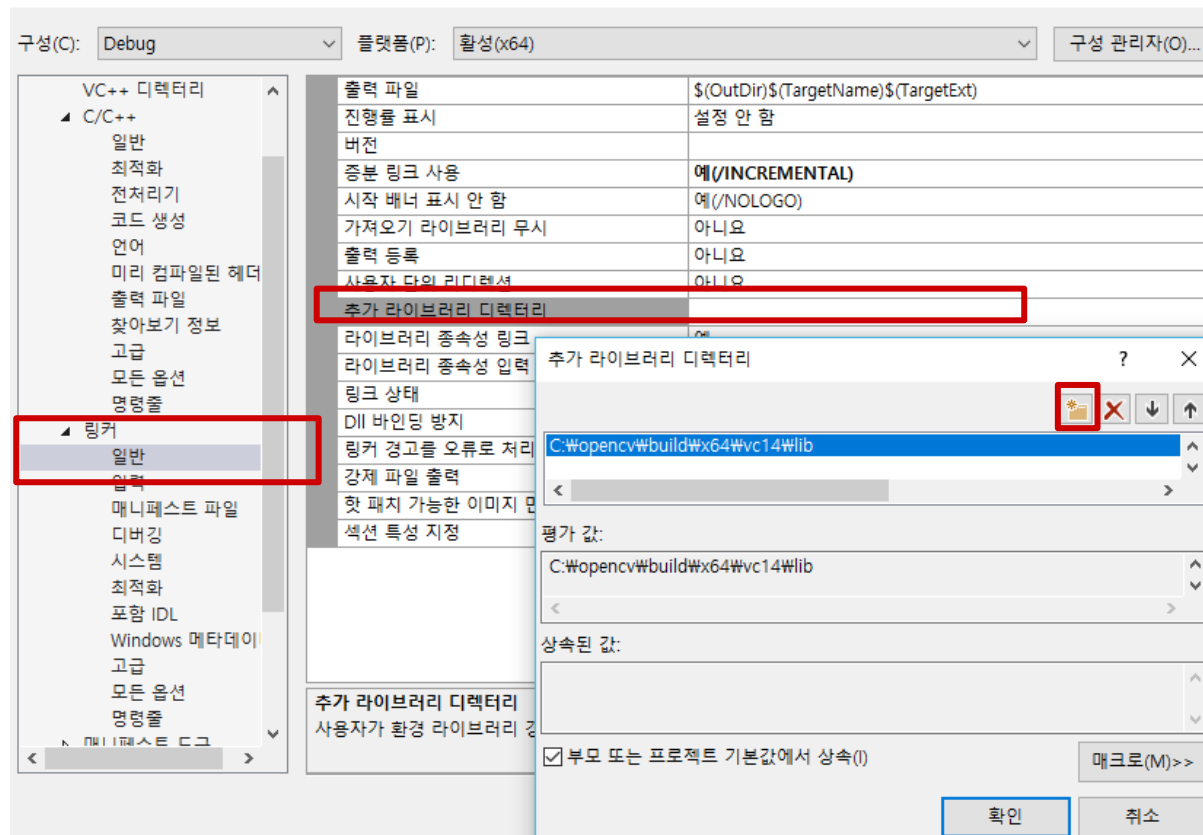
- 프로젝트 – 프로젝트 속성 – 구성 속성 – C/C++ – 일반
- 추가 포함 디렉터리 – Header File Directory 지정
- C:\w에 압축 해제한 경우 C:\wopencv\build\include 를 추가



# Install & Setting

## □ Setting

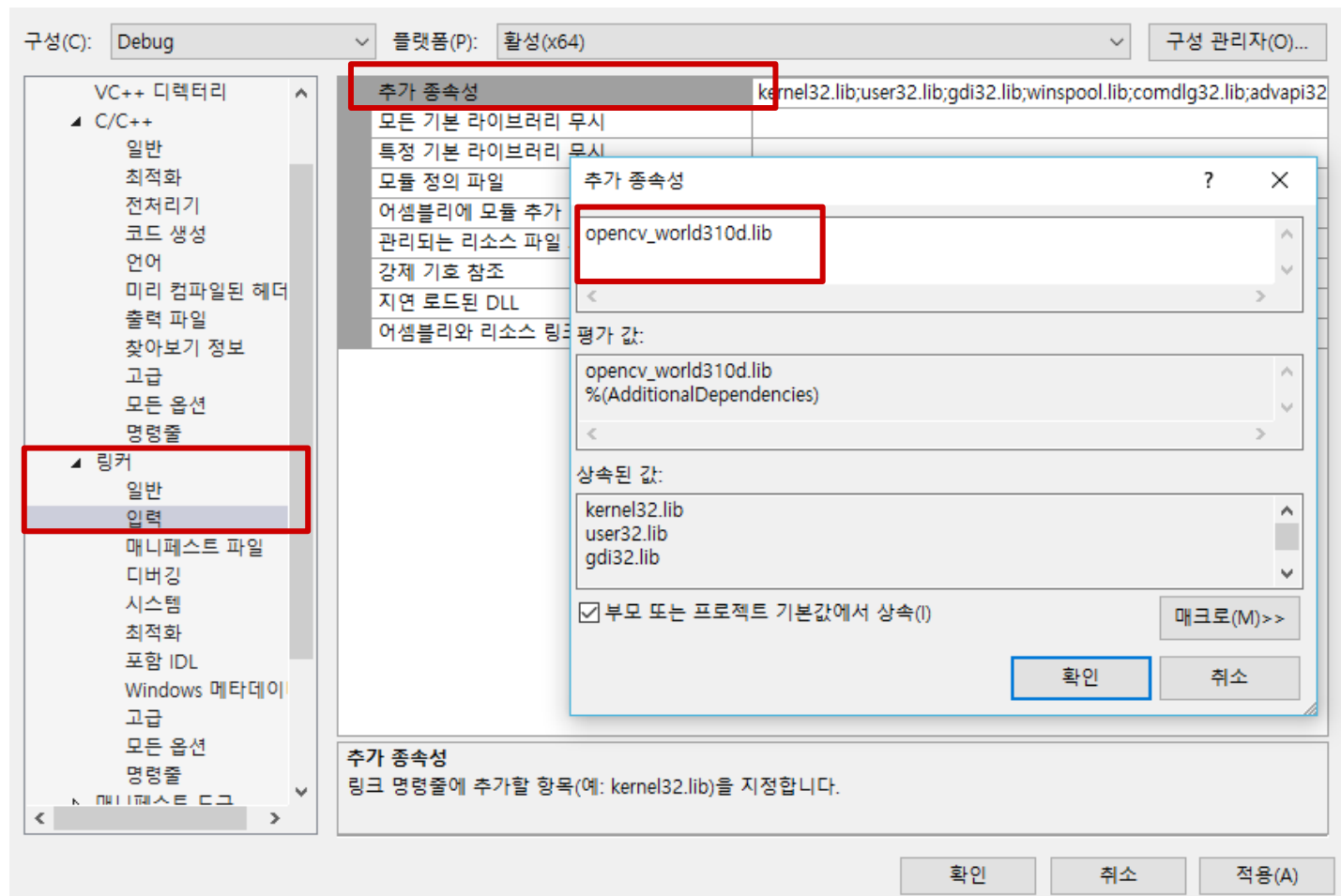
- 링커 – 일반 – 추가 라이브러리 디렉터리 – Library File Directory 지정
  - (64bit) C:\wopencv\build\x64\vc14\lib 추가



# Install & Setting

## □ Setting

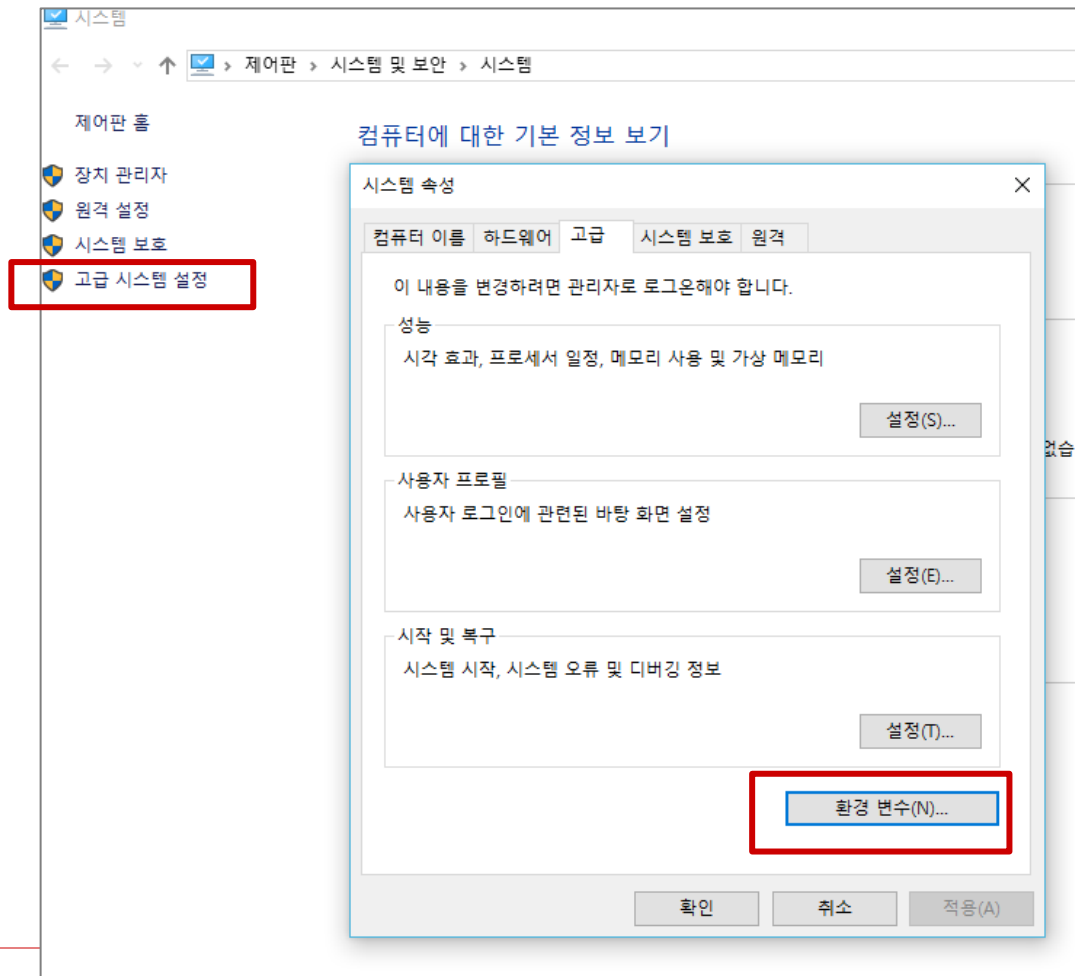
### ■ 입력 - 추가 종속성



# Install & Setting

## □ Setting

### ■ 컴퓨터 우클릭 – 속성 – 고급 시스템 설정 – 환경 변수



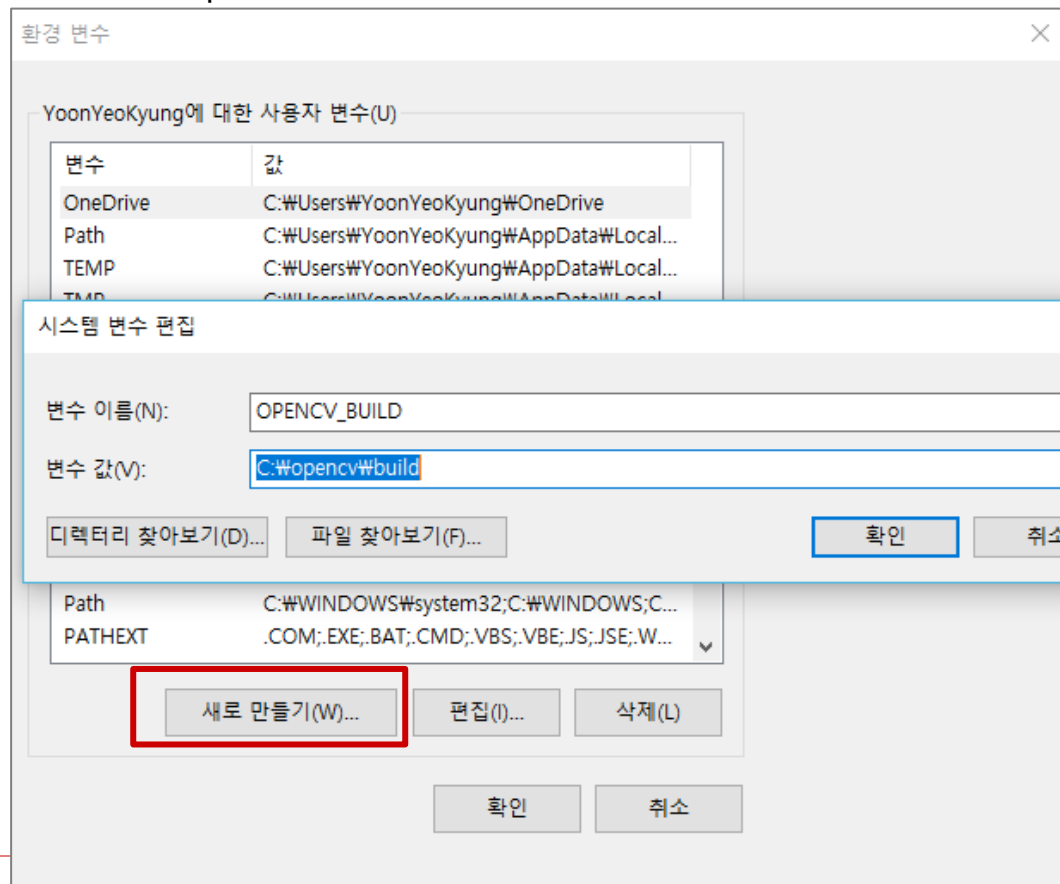
# Install & Setting

## □ Setting

### ■ 시스템 변수 목록 - 새로 만들기

□ 변수 이름 : OPENCV\_BUILD

□ 변수 값 : C:\wopencv\build

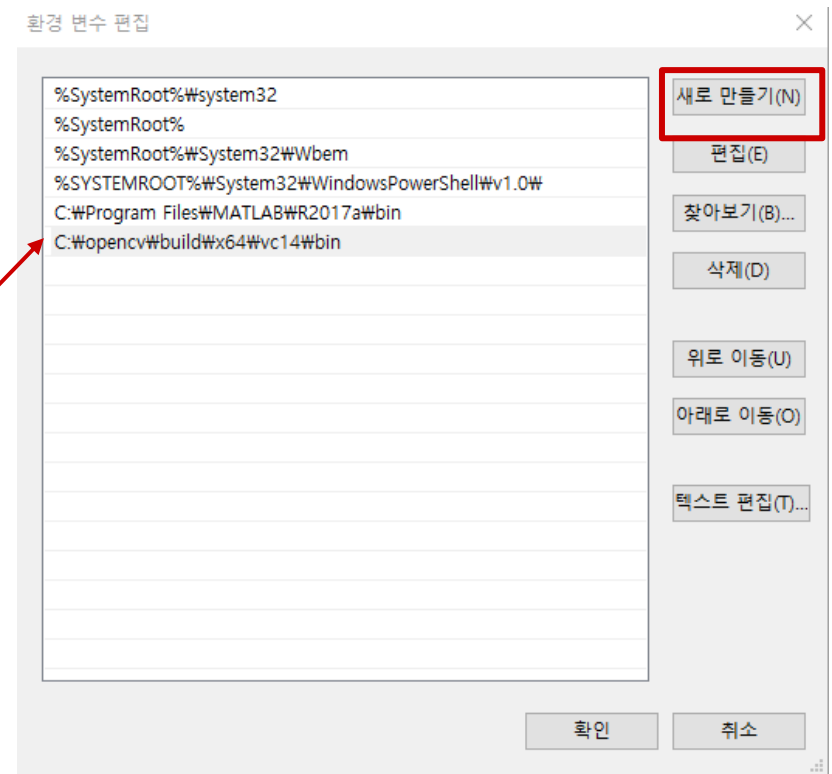
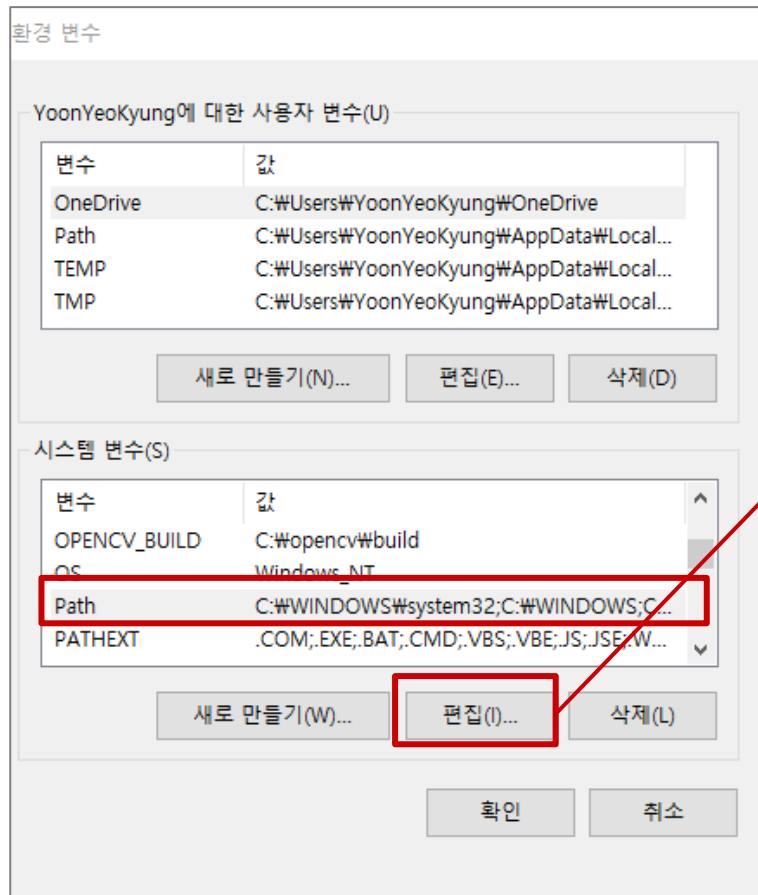




# Install & Setting

## □ Setting

- 시스템 변수 목록 - Path - 편집
- 새로 만들기 - C:\wopencv\build\x64\vc14\bin 추가



# Image I/O using OpenCV

---

# Image I/O using OpenCV

## □ Mat Class : `class CV_EXPORTS Mat`

### ■ OpenCV 에서 사용되는 클래스

- 1채널 또는 다채널의 실수, 복소수, 행렬, 영상 등의 수치 데이터를 표현하는 N차원 행렬 클래스

### ■ 주요 생성자

#### □ `Mat();`

- Default

#### □ `Mat(int rows, int cols, int type);`

- rows : 영상의 높이, cols : 영상의 폭, type : 영상의 Data type & Channel

#### □ `Mat(Size size, int type);`

- size : 영상의 폭과 높이를 저장하는 구조체
- Ex) `Mat(Size(cols, rows), type);`

# Image I/O using OpenCV

## □ Data Type & Channel Pattern

CV\_<bit\_depth>{U | S | F}C(<number\_of\_channel>)

Data Type : unsigned | signed | float

CV\_8UC1 : 8 비트 픽셀/부호가 없는 정수/1채널

CV\_8UC3 : 8 비트 픽셀/부호가 없는 정수/3채널

CV\_16SC3 : 16 비트 픽셀/부호가 있는 정수/3채널

CV\_32FC4 : 32 비트 픽셀/부동 소수점 방식/4채널

# Image I/O using OpenCV

## □ Namespace cv

- OpenCV의 모든 함수와 클래스를 포함하고 있음
- 헤더파일 다음에 'using namespace cv' 문을 추가하여 사용
- 주요 메서드

□ `Mat imread( const String& filename, int flags = IMREAD_COLOR );`

- 영상 파일을 불러들여요
- 지원 파일 포맷 :

□ `bool imwrite( const String& filename, const std::vector<int>& pa`

- 객체에 저장된 영상정보

□ `void imshow( const String& winname, InputArray mat );`

- Mat 객체를 특정 윈도우에 Display

□ `int waitKey( int delay = 0 );`

- 지정된 시간만큼 키의 입력을 기다리는 메서드
- 0을 입력할 경우 키보드 입력이 들어올 때 까지 무한 대기

Currently, the following file formats are supported:

- Windows bitmaps - `*.bmp`, `*.dib` (always supported)
- JPEG files - `*.jpeg`, `*.jpg`, `*.jpe` (see the *\*Notes\** section)
- JPEG 2000 files - `*.jp2` (see the *\*Notes\** section)
- Portable Network Graphics - `*.png` (see the *\*Notes\** section)
- WebP - `*.webp` (see the *\*Notes\** section)
- Portable image format - `*.pbm`, `*.pgm`, `*.ppm`, `*.pxm`, `*.pnm` (always supported)
- Sun rasters - `*.sr`, `*.ras` (always supported)
- TIFF files - `*.tiff`, `*.tif` (see the *\*Notes\** section)
- OpenEXR image files - `*.exr` (see the *\*Notes\** section)

# Image I/O using OpenCV

## □ Image I/O Test

- Test Set : Lena.jpg
  - Image Processing의 표준 영상으로 많이 사용됨
  - row(Height) : 512, col(Width) : 512



# Image I/O using OpenCV

## □ Image I/O Test

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

using namespace cv;

void main()
{
    // 현재 프로젝트 폴더에 있는 'lena.jpg' 이미지 파일을 불러온 후, Image 변수에 저장
    Mat Image = imread("lena.jpg", IMREAD_COLOR);

    // 윈도우 이름 설정
    namedWindow("display", WINDOW_AUTOSIZE);

    // imshow("윈도우 이름", 윈도우에 디스플레이 하고자 하는 이미지)
    imshow("display", Image);

    // imwrite("저장하고자 하는 이미지 파일 이름.파일확장자", 저장하고자 하는 이미지)
    imwrite("lena_new.jpg", Image);

    waitKey(0);
}
```

# Video I/O using OpenCV

---



# Video I/O using OpenCV

## □ Video class

### ■ VideoCapture, VideoWriter

- VideoCapture : 비디오 파일을 열고, 비디오 스트림을 읽기 위한 클래스
- VideoWriter : 비디오를 작성하기 위한 클래스

### ■ 주요 생성자

- VideoCapture(); VideoWriter();
  - Default
- VideoCapture(const string & filename);
  - filename : 비디오 파일 이름(video.avi) or 이미지 시퀀스(img\_00.jpg, img\_01.jpg, img\_02.jpg) or 비디오 스트림 URL
- VideoWriter(const string & filename, int fourcc, double fps, Size frameSize, bool isColor=true)
  - filename : 출력 비디오 파일 이름
  - fourcc : 프레임을 압축하는데 사용되는 4자의 코덱  
(코덱 목록 : <http://www.fourcc.org/codecs.php>)
  - fps : 만들어진 비디오 스트림의 프레임 속도
  - frameSize : 비디오 프레임의 크기
  - isColor : 0이 아닌 경우 컬러 프레임, 그렇지 않으면 그레이 스케일 프레임으로 인코딩

# Video I/O using OpenCV

## □ Video I/O Test

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

using namespace cv;

void main()
{
    // 프로젝트 폴더에 있는 'video.mp4' 비디오 파일을 불러옴
    VideoCapture capture("video.mp4");
    // 프레임을 저장할 Mat 선언
    Mat frame;

    // 비디오를 불러오지 못했을 때 에러 메시지 출력 후 종료
    if (!capture.isOpened())
    {
        printf("Video file can not open.\n");
        return;
    }

    // 출력 창 이름을 정함
    namedWindow("video");

    while (1)
    {
        // 캡처한 비디오 프레임을 Mat에 저장
        capture >> frame;

        // 비디오가 끝났을 때
        if (frame.empty())
            break;

        // 출력 창에 캡처한 프레임 출력
        imshow("video", frame);
        if (waitKey(10) > 0)
            break;
    }
}
```

# Video I/O using OpenCV

## □ Video I/O Test

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

using namespace std;
using namespace cv;

void main()
{
    // 프로젝트 폴더에 있는 'video.mp4' 비디오 파일을 불러옴
    VideoCapture capture("video.mp4");

    if (!capture.isOpened())
    {
        printf("Video file can not open.\n");
        return;
    }
    VideoWriter writer;

    // 비디오 프레임들 저장할 Mat 벡터 선언
    vector<Mat> video;

    // VideoWriter(파일 이름, 코덱, 프레임 속도, 비디오 크기, 컬러);
    writer.open("NewVideo.avi", CV_FOURCC('D', 'I', 'V', 'X'), 30, Size(1920, 1080), true);

    while (1)
    {
        // 캡처한 프레임을 Mat에 저장
        Mat frame;
        capture >> frame;

        // 비디오가 끝날 때까지
        if (frame.empty())
            break;

        // 저장한 프레임을 벡터에 저장
        video.push_back(frame);
    }

    // 저장된 벡터의 비디오 프레임들 뒤에서부터 새로 저장할 비디오에 입력
    for (int i = 0; i < video.size(); i++)
        writer.write(video[video.size() - (i + 1)]);
}
```