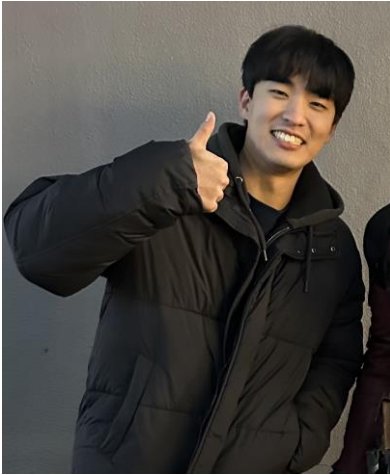


# Open CV

## Open Source Computer Vision

Department of Electrical Engineering, Incheon National University  
**Chulmin Jeong, Youngkeun Kim**

# 강사 소개



## ■ 김영근 (Youngkeun Kim)

■ ~2024: 인천대학교 전기공학과 학사

■ 2024~현재: 인천대학교 전기공학과 석박사 통합과정 재학 중

■ 무인지능 시스템제어 연구실 소속

(지도교수: 강창묵 교수, <https://uniconlab.wixsite.com/main>)

■ 관심분야: 고장진단, 자율주행, SLAM, 인공지능

■ 적용플랫폼: 자율주행차량, 모바일로봇, 로봇 팔, 4족보행로봇 등

■ Projects:

- CNN 기반의 wavelet 이미지 분류
- 인천공항 폐기물 이송 로봇 개발
- ROS 기반 시뮬레이션 환경 구성
- ...

# Contents

## 1. Open CV 개요

- 1) Open CV란?
- 2) Open CV 설치 (Python 기반)
- 3) Open CV의 주요기능
- 4) CV bridge 기반 ROS 환경 및 Open CV 연동

## 2. Open CV 실습

- 1) 이미지 출력
- 2) 이미지 gray scale
- 3) 이미지 edge 검출
- 4) 이미지 회전/크기 조절/기울이기
- 5) 이미지 filtering
- 6) 차선 인식 (Hough / HSV / Hough+HSV)
- 7) LIMO 카메라 사용

# 1. Open CV 개요

# 1. Open CV 개요

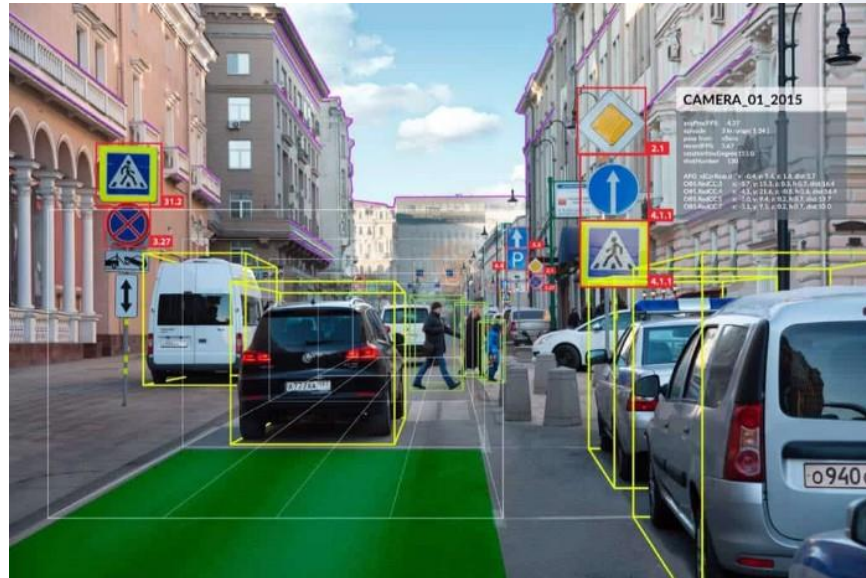
## 1) Open CV(Open Source Computer Vision)란?



- Open CV는 “**Open Source Computer Vision Library**”의 약자로, 이미지 및 동영상 처리에 사용되는 라이브러리
- C++, Python, Java 등 다양한 프로그래밍 언어를 지원하며, 이미지 프로세싱, 컴퓨터 비전 및 머신러닝 알고리즘에 대한 다양한 함수와 라이브러리 제공
- Open CV는 무료로 제공되며, 비즈니스 및 개인 프로젝트 모두에서 사용 가능

# 1. Open CV 개요

## 1) Open CV(Open Source Computer Vision)란?

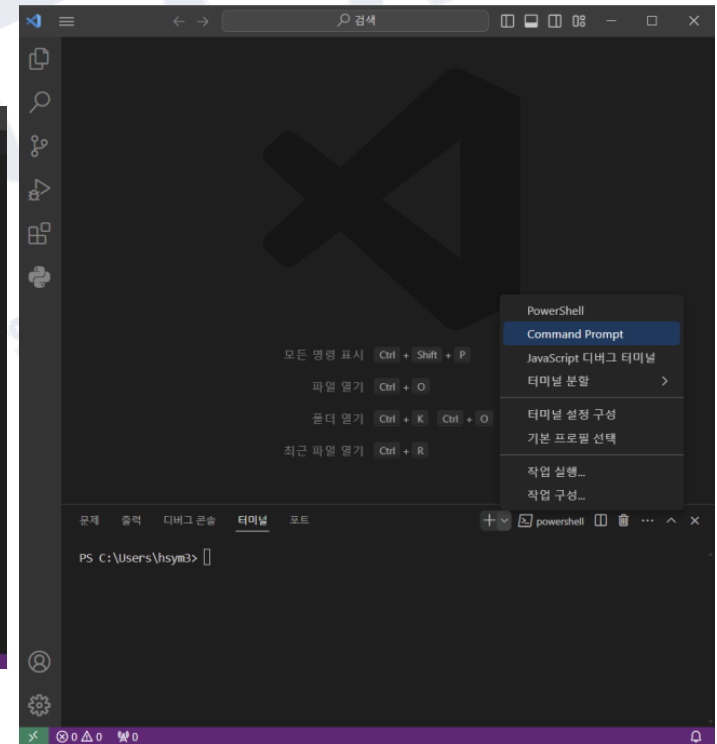
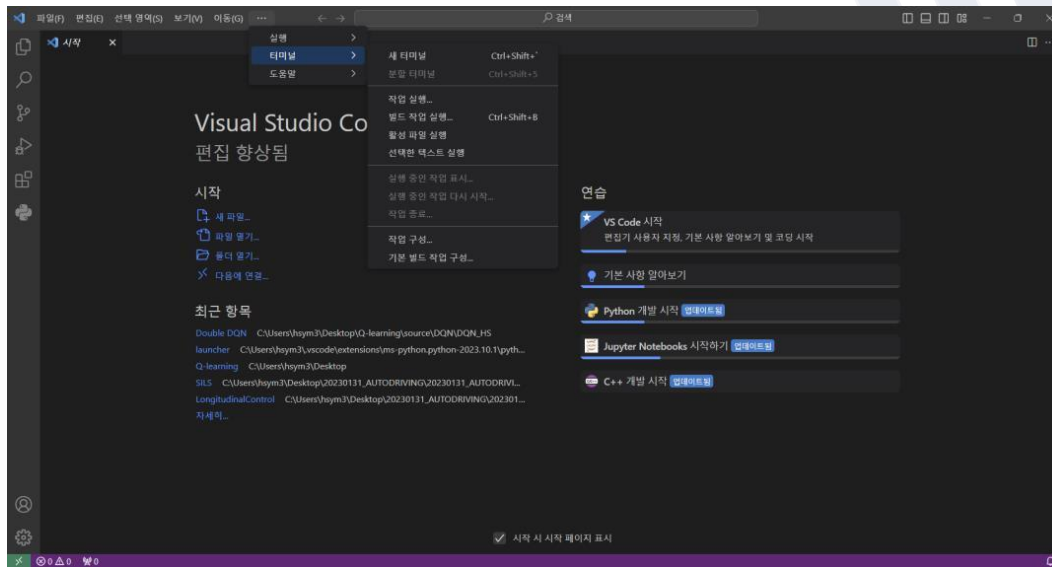


- Open CV를 통해 이미지나 영상에서 **객체 검출, 추적, 특징 추출, 패턴 인식** 등의 작업 수행 가능
- Open CV에서는 이미지 필터링, 색 공간 변환, 이미지 모핑, 카메라 보정 등 다양한 기능을 제공
- 최근 **Open CV**와 **Deep Learning**을 결합하여 이미지 및 영상 데이터를 쉽게 처리할 수 있게 되었으며, 이를 기반으로 객체 인식 및 추적 등의 작업을 빠르게 수행할 수 있게 되었음

## 1. Open CV 개요

## 2) Open CV 설치 (Python 기반)

➔ VS Code 및 Python3는 사전에 설치되어 있다고 가정



- VS Code에서 터미널을 실행 후, 명령 프롬프트(Command Prompt) 클릭

# 1. Open CV 개요

## 2) Open CV 설치 (Python 기반)

\$ pip install opencv-python

\$ pip install numpy

```

PS C:\Users\USER> pip install opencv-python
Collecting opencv-python
  Downloading opencv_python-4.10.0.84-cp37-abi3-win_amd64.whl.metadata (20 kB)
Collecting numpy>=1.21.2 (from opencv-python)
  Downloading numpy-2.0.0-cp312-cp312-win_amd64.whl.metadata (60 kB)
    ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 60.9/60.9 kB 3.2 MB/s eta 0:00:00
  Downloading opencv_python-4.10.0.84-cp37-abi3-win_amd64.whl (38.8 MB)
    ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 38.8/38.8 MB 11.3 MB/s eta 0:00:00
  Downloading numpy-2.0.0-cp312-cp312-win_amd64.whl (16.2 MB)
    ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 16.2/16.2 MB 11.7 MB/s eta 0:00:00
Installing collected packages: numpy, opencv-python
Successfully installed numpy-2.0.0 opencv-python-4.10.0.84

[notice] A new release of pip is available: 24.0 -> 24.1.1
[notice] To update, run: python.exe -m pip install --upgrade pip
PS C:\Users\USER> pip install numpy
Requirement already satisfied: numpy in c:\users\user\appdata\local\programs\python\python312\lib\site-packages (2.0.0)

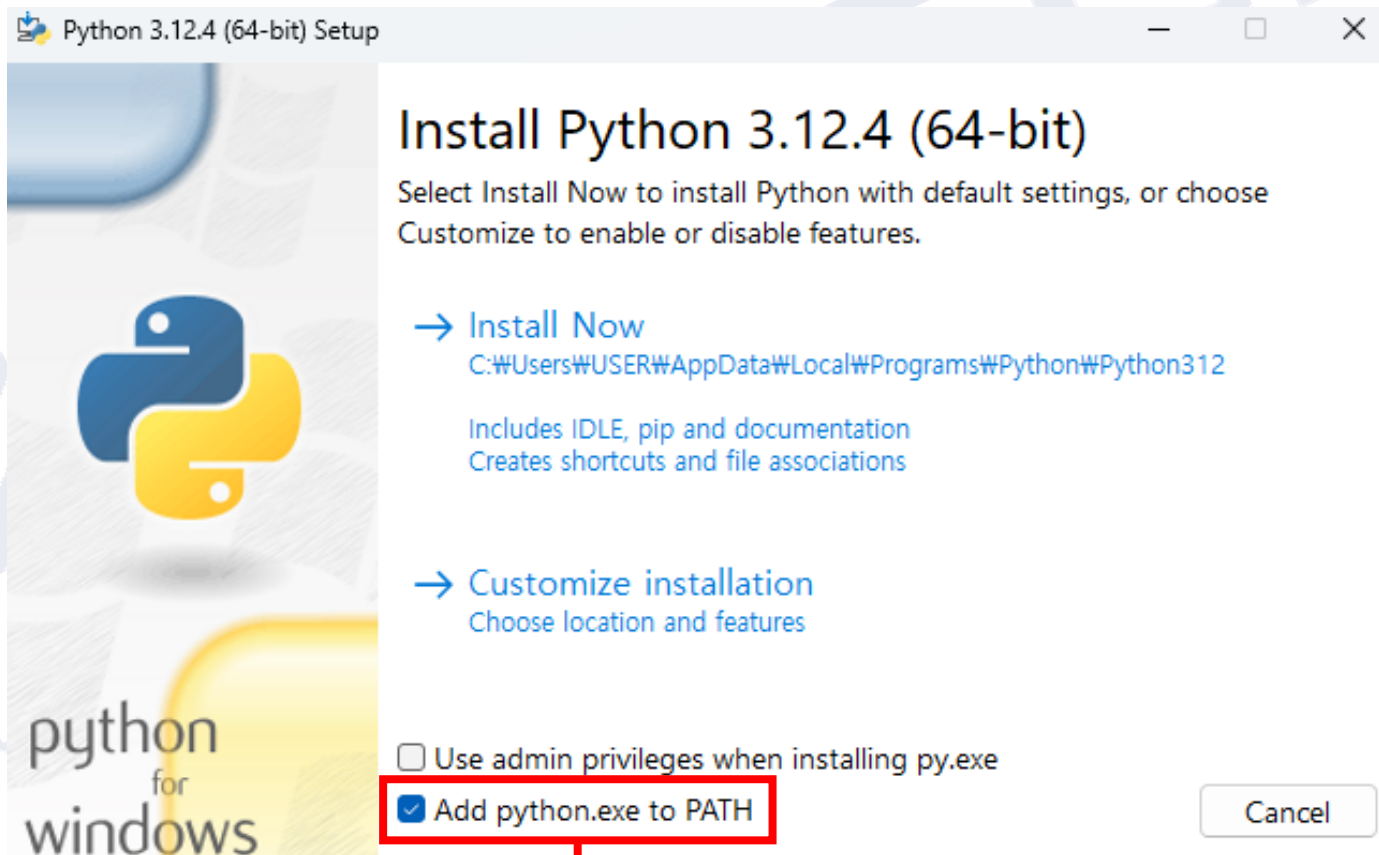
[notice] A new release of pip is available: 24.0 -> 24.1.1
[notice] To update, run: python.exe -m pip install --upgrade pip
  
```



# 1. Open CV 개요

## 2) Open CV 설치 (Python 기반)

➔ 만약 설치 과정에서 pip 관련 오류가 발생한다면, Python 재설치 수행

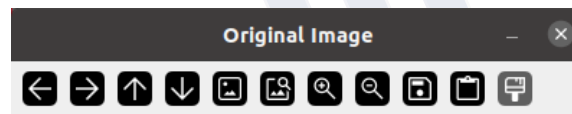


➔ Python 재설치 시, 반드시 해당 부분 체크 !

# 1. Open CV 개요

## 3) Open CV 주요 기능

- 이미지 읽기 및 쓰기  
→ 이미지를 파일에서 읽어오고, 처리된 이미지를 파일로 저장 가능



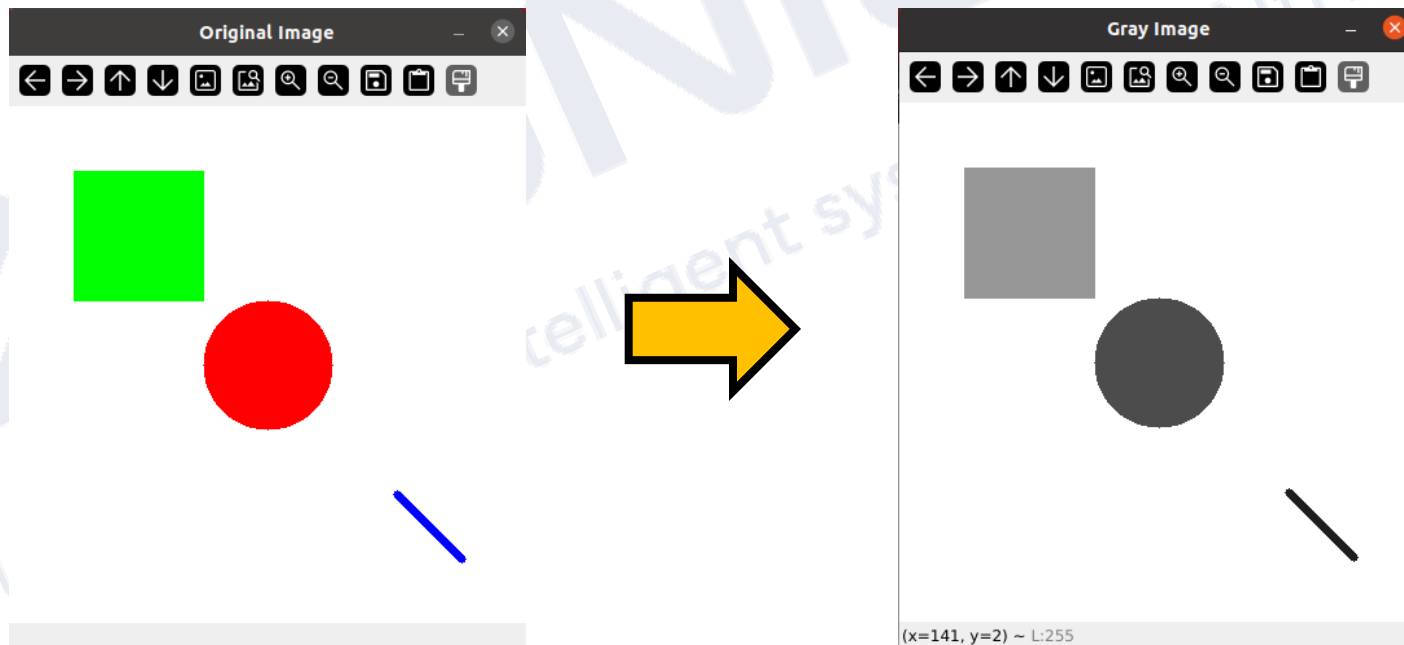
# 1. Open CV 개요

## 3) Open CV 주요 기능

- 이미지 변환

→ 이미지를 다양한 색상 공간으로 변환 가능

→ 다음 예시와 같이, RGB 이미지를 Gray scale 또는 HSV 색상 공간으로 변환 가능



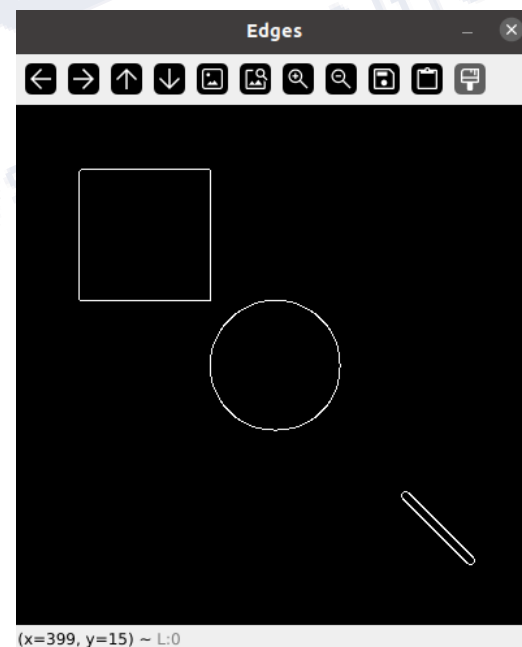
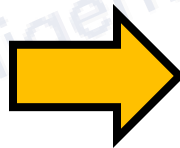
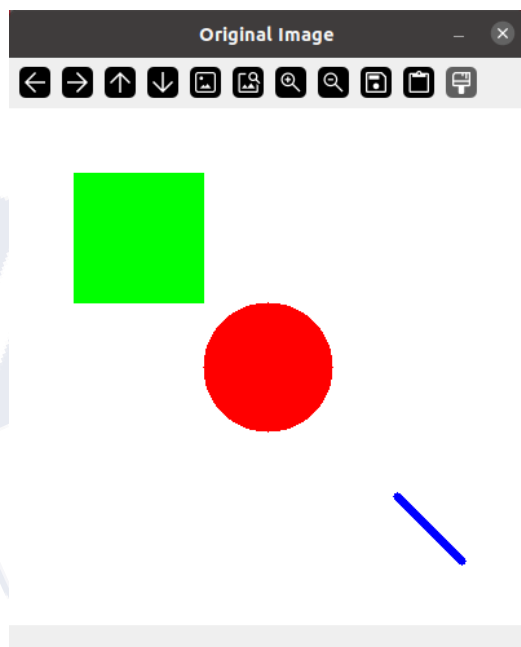
# 1. Open CV 개요

## 3) Open CV 주요 기능

- Edge 검출

→ 이미지에서 객체의 경계를 찾기 위해 edge를 검출

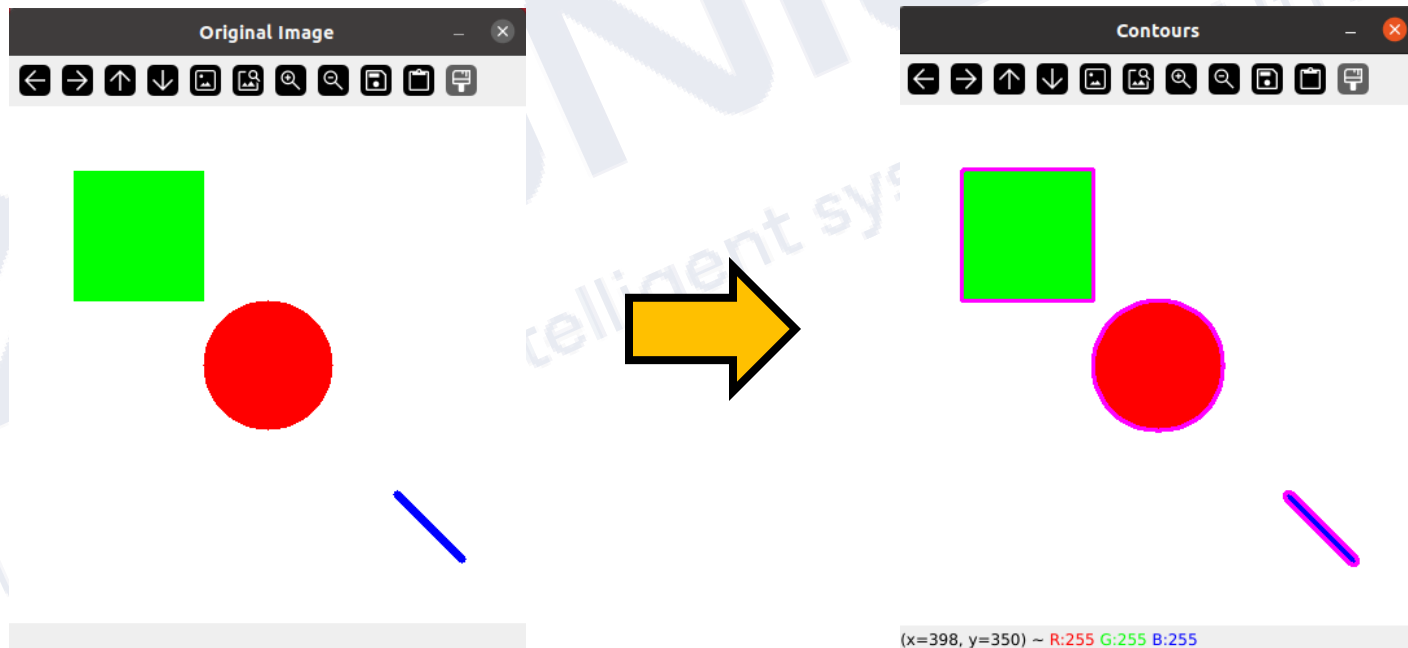
→ 대표적인 edge 검출 기법으로 'Canny edge' 검출 기법이 존재



# 1. Open CV 개요

## 3) Open CV 주요 기능

- 윤곽선 검출  
→ 이미지에서 객체의 윤곽선을 검출



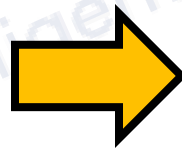
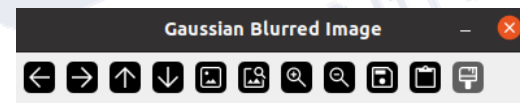
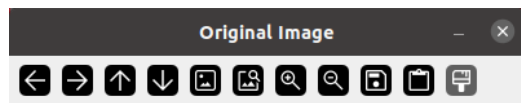
# 1. Open CV 개요

## 3) Open CV 주요 기능

- 이미지 필터링

→ 이미지를 필터링하여 노이즈를 제거하거나 부드럽게 변환 가능

→ 대표적인 필터링 기법으로 'Gaussian Blur', 'Median Blur' 등이 있음



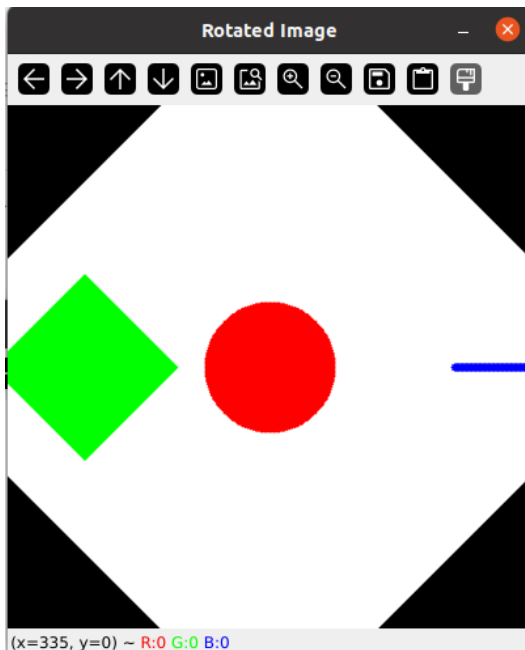
(x=160, y=1) ~ R:255 G:255 B:255

# 1. Open CV 개요

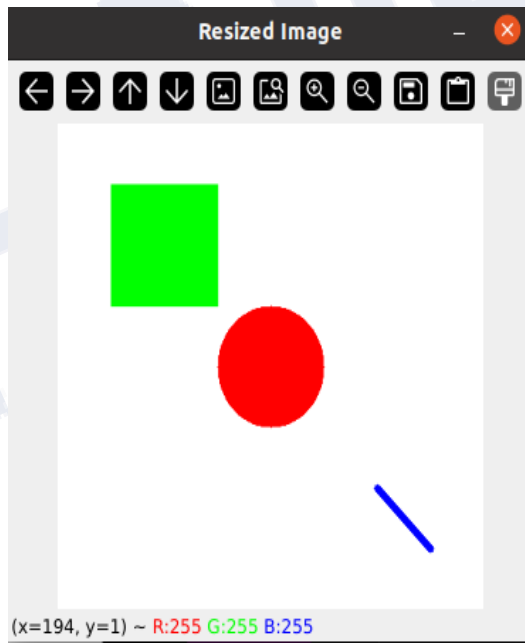
## 3) Open CV 주요 기능

- 이미지 조정

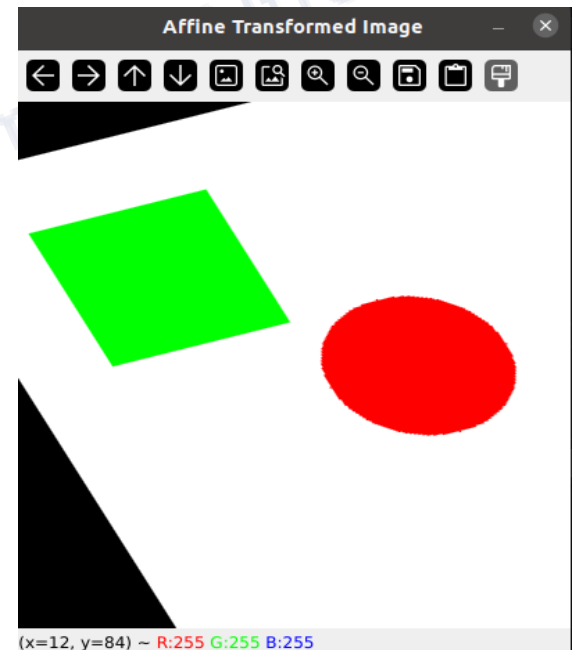
→ 이미지 회전, 크기 조절, 기울이기 등의 기능을 통해 이미지 데이터를 변환 가능



이미지 회전



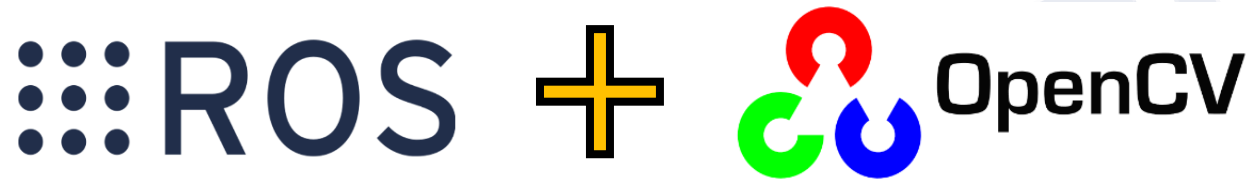
이미지 크기 조절



이미지 기울이기

# 1. Open CV 개요

## 4) CV bridge 기반 ROS 환경 및 Open CV 연동



- 실시간 이미지 처리

- 로봇에 부착된 Camera 센서를 통해 수집된 이미지를 실시간으로 처리하여 장애물 감지, 객체 추종 등 다양한 작업을 수행 가능

- 모듈성 및 재사용성

- ROS(Robot Operating System)의 Module(모듈)식 구조를 활용하여 Open CV 기반 이미지 처리 기능을 독립적인 노드로 구현하고, 다양한 프로젝트에서 재사용 가능

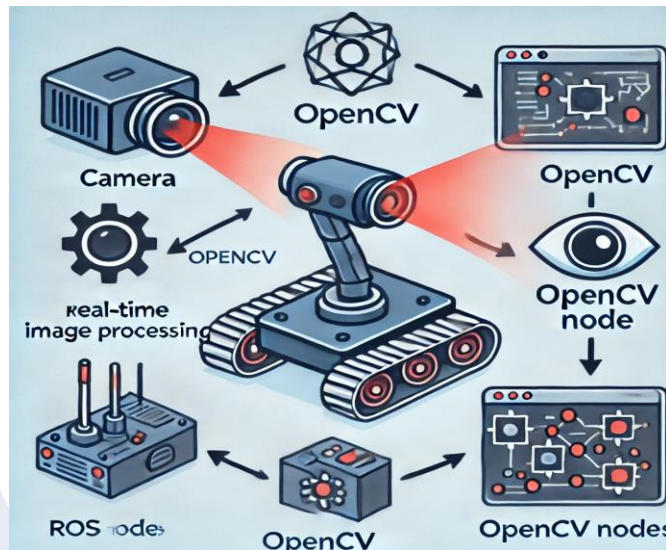
- 커뮤니티 및 오픈 소스 지원

- ROS와 Open CV는 모두 오픈 소스 프로젝트로, 광범위한 커뮤니티 지원과 수 많은 자료를 제공
- 최신 기술에서 발생할 수 있는 문제에 대한 해결이 용이하며, 이를 바탕으로 다양한 최신 기술을 편하게 사용 가능



# 1. Open CV 개요

## 4) CV bridge 기반 ROS 환경 및 Open CV 연동



- 다양한 센서 데이터 통합

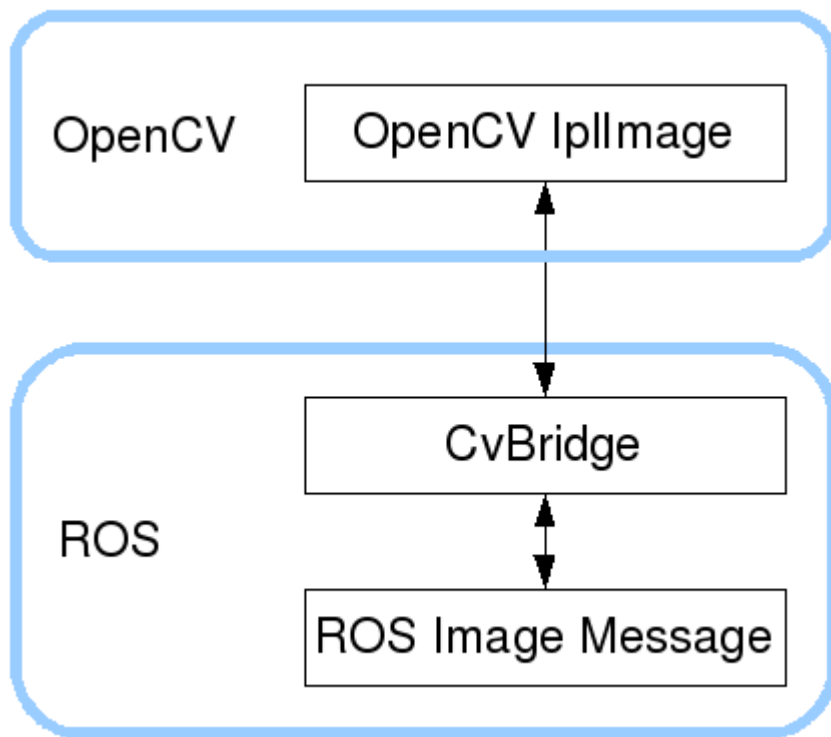
→ Open CV를 통해 처리한 이미지 데이터와 더불어, 다양한 센서들을 결합하여 도출한 데이터를 바탕으로 더욱 정밀하게 환경 인식 수행 가능

- 시뮬레이션 및 로봇 적용

→ Gazebo 시뮬레이터를 통해 가상 환경에서 로봇을 테스트한 뒤, 실제 로봇에 적용함으로써 테스트 및 개발 시간을 단축 가능

# 1. Open CV 개요

## 4) CV bridge 기반 ROS 환경 및 Open CV 연동



### ● cv\_bridge란?

→ ROS에서 제공하는 패키지로, ROS 이미지 메시지와 Open CV 이미지 간의 원활한 변환을 돕는 기능

→ cv\_bridge를 통해 ROS 환경의 노드가 수신한 이미지를 Open CV 형식으로 변환하여, 다양한 이미지 처리 작업을 수행 가능

# 1. Open CV 개요

## 4) CV bridge 기반 ROS 환경 및 Open CV 연동

- ROS noetic 환경 기반 Open CV 설치

→ \$ pip install opencv-python ( 또는 \$ pip3 install opencv-python )

```
gihoon@gihoon-B650M-HDV-M-2:~$ pip install opencv-python
Collecting opencv-python
  Downloading opencv_python-4.10.0.84-cp37-abi3-manylinux_2_17_x86_64.manylinux2
014_x86_64.whl (62.5 MB)
    | 62.5 MB 11.8 MB/s
Requirement already satisfied: numpy>=1.17.3; python_version >= "3.8" in ./.loca
l/lib/python3.8/site-packages (from opencv-python) (1.21.5)
Installing collected packages: opencv-python
Successfully installed opencv-python-4.10.0.84
```

- ROS noetic 환경에 Open CV 설치가 완료되었는지 확인

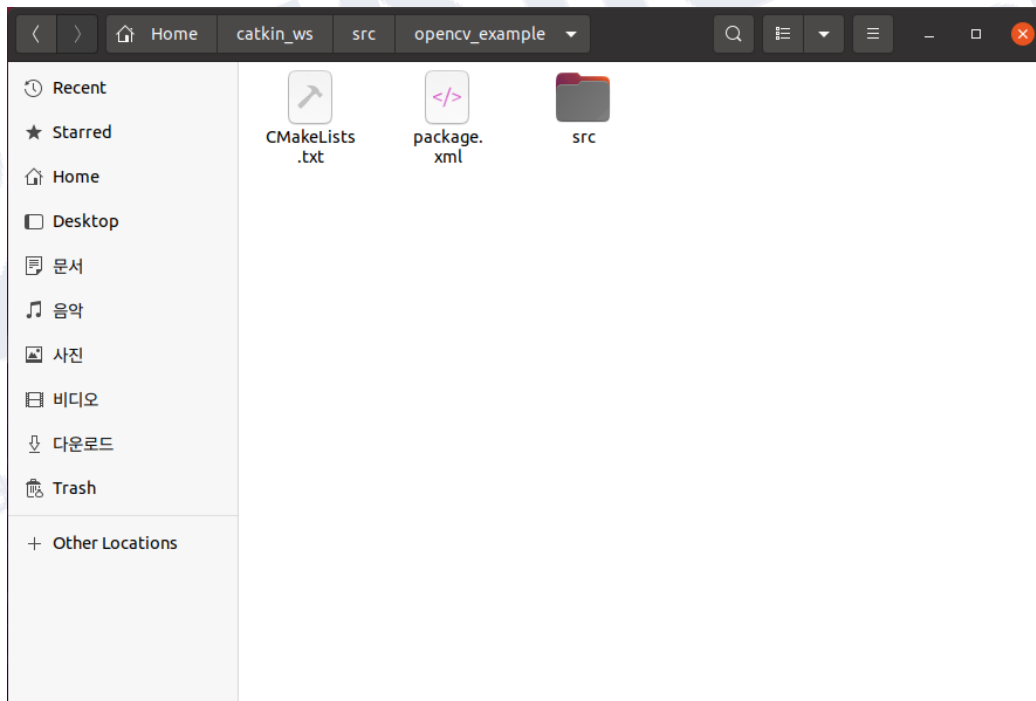
→ \$ dpkg -l | grep ros-noetic-cv-bridge

```
gihoon@gihoon-B650M-HDV-M-2:~$ dpkg -l | grep libopencv
ii  libopencv-calib3d-dev:amd64      4.2.0+dfsg-5      amd64
ii  libopencv-calib3d4.2:amd64      4.2.0+dfsg-5      amd64
ii  libopencv-contrib-dev:amd64     4.2.0+dfsg-5      amd64
ii  libopencv-contrib4.2:amd64      4.2.0+dfsg-5      amd64
ii  libopencv-core-dev:amd64        4.2.0+dfsg-5      amd64
ii  libopencv-core4.2:amd64         4.2.0+dfsg-5      amd64
ii  libopencv-dev                   4.2.0+dfsg-5      amd64
ii  libopencv-dnn-dev:amd64         4.2.0+dfsg-5      amd64
ii  libopencv-dnn4.2:amd64          4.2.0+dfsg-5      amd64
ii  libopencv-features2d-dev:amd64  4.2.0+dfsg-5      amd64
ii  libopencv-features2d4.2:amd64   4.2.0+dfsg-5      amd64
ii  libopencv-flann-dev:amd64       4.2.0+dfsg-5      amd64
```

# 1. Open CV 개요

## 4) CV bridge 기반 ROS 환경 및 Open CV 연동

- ROS noetic 환경에서 Camera 이미지 수신 및 Open CV 기반 처리 실습 (1/7)
  - `$ cd ~/catkin_ws/src`
  - `$ catkin_create_pkg opencv_example rospy std_msgs sensor_msgs cv_bridge`



# 1. Open CV 개요

## 4) CV bridge 기반 ROS 환경 및 Open CV 연동

- ROS noetic 환경에서 Camera 이미지 수신 및 Open CV 기반 처리 실습 (2/7)
  - \$ cd opencv\_example (새롭게 생성한 패키지로 이동)
  - \$ gedit CMakeLists.txt (CMakeLists 파일을 수정 → cv\_bridge와 Open CV 의존성 추가)



```
1 cmake_minimum_required(VERSION 3.0.2)
2 project(opencv_example)
3
4 ## Compile as C++11, supported in ROS Kinetic and newer
5 # add_compile_options(-std=c++11)
6
7 ## Find catkin macros and libraries
8 ## if COMPONENTS list like find_package(catkin REQUIRED COMPONENTS xyz)
9 ## is used, also find other catkin packages
10 find_package(catkin REQUIRED COMPONENTS
11   rospy
12   std_msgs
13   sensor_msgs
14   cv_bridge
15 )
16
17 find_package(OpenCV REQUIRED)
18
19 catkin_package()
20
21 include_directories(
22   ${catkin_INCLUDE_DIRS}
23   ${OpenCV_INCLUDE_DIRS}
24 )
```

# 1. Open CV 개요

## 4) CV bridge 기반 ROS 환경 및 Open CV 연동

- ROS noetic 환경에서 Camera 이미지 수신 및 Open CV 기반 처리 실습 (3/7)
  - \$ cd opencv\_example (새롭게 생성한 패키지로 이동)
  - \$ gedit package.xml (package.xml 파일을 수정 → cv\_bridge와 Open CV 의존성 추가)

```
51 <buildtool_depend>catkin</buildtool_depend>
52 <build_depend>cv_bridge</build_depend>
53 <build_depend>rospy</build_depend>
54 <build_depend>sensor_msgs</build_depend>
55 <build_depend>std_msgs</build_depend>
56 <build_export_depend>cv_bridge</build_export_depend>
57 <build_export_depend>rospy</build_export_depend>
58 <build_export_depend>sensor_msgs</build_export_depend>
59 <build_export_depend>std_msgs</build_export_depend>
60 <exec_depend>cv_bridge</exec_depend>
61 <exec_depend>rospy</exec_depend>
62 <exec_depend>sensor_msgs</exec_depend>
63 <exec_depend>std_msgs</exec_depend>
64
```



```
<build_depend>cv_bridge</build_depend>
<exec_depend>cv_bridge</exec_depend>

<build_depend>opencv</build_depend>
<exec_depend>opencv</exec_depend>
```

# 1. Open CV 개요

## 4) CV bridge 기반 ROS 환경 및 Open CV 연동

- ROS noetic 환경에서 Camera 이미지 수신 및 Open CV 기반 처리 실습 (4/7)
  - \$ cd ~/catkin\_ws/src/opencv\_example/ (새롭게 생성한 패키지 내 src 폴더로 이동)
  - \$ gedit test.py (src 폴더 내 python 파일 생성 후 저장)
  - \$ chmod +x test.py (새롭게 생성한 python 파일에 실행 권한 부여)

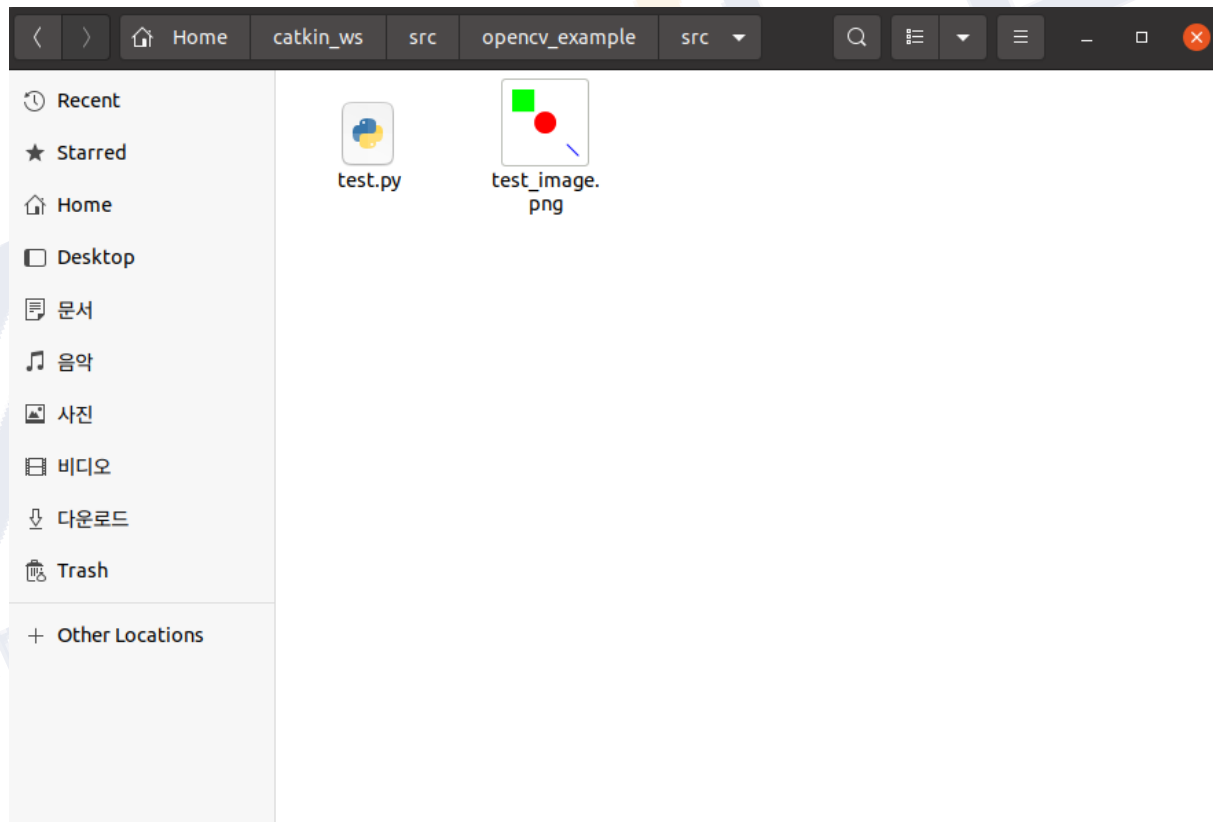
```
gihoon@gihoon-B650M-HDV-M-2:~$ cd catkin_ws/src/opencv_example/src/  
gihoon@gihoon-B650M-HDV-M-2:~/catkin_ws/src/opencv_example/src$ chmod +x test.py
```

- \$ cd ~/catkin\_ws/ && catkin\_make (catkin\_ws 빌드)
- \$ source devel/setup.bash (빌드 후, 환경 변수 재 설정)

# 1. Open CV 개요

## 4) CV bridge 기반 ROS 환경 및 Open CV 연동

- ROS noetic 환경에서 Camera 이미지 수신 및 Open CV 기반 처리 실습 (5/7)  
→ Open CV 실습에 사용할 이미지를 다운로드 후, 저장





# 1. Open CV 개요

## 4) CV bridge 기반 ROS 환경 및 Open CV 연동

- ROS noetic 환경에서 Camera 이미지 수신 및 Open CV 기반 처리 실습 (6/7)
  - \$ cd ~/catkin\_ws/src/opencv\_example/src/ (새롭게 생성한 패키지 내 src 폴더 진입)
  - \$ gedit test.py (새롭게 생성한 test.py 파일에 아래와 같이 작성 후, 저장)

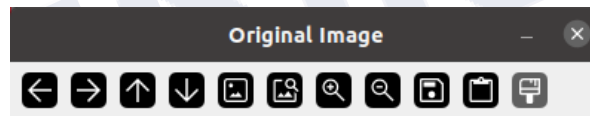
```
1  #!/usr/bin/env python3
2  import cv2
3  import rospy
4
5
6  image_path = '/home/user/catkin_ws/src/opencv_example/src/test_image.png' # 이미지 경로를 설정하세요
7  image = cv2.imread(image_path)
8
9  # 1. 원본 이미지 표시
10 cv2.imshow('Original Image', image)
11
12 cv2.waitKey(0)
13 cv2.destroyAllWindows()
14 |
```

→ 본인이 저장한 이미지 파일의 경로를 작성 !!

# 1. Open CV 개요

## 4) CV bridge 기반 ROS 환경 및 Open CV 연동

- ROS noetic 환경에서 Camera 이미지 수신 및 Open CV 기반 처리 실습 (7/7)
  - `$ roscore` (첫 번째 터미널)
  - `$ cd ~/catkin_ws/src/opencv_example/src/` (두 번째 터미널)
  - `$ python3 test.py` (두 번째 터미널)



## 2. Open CV 실습

git clone [https://github.com/0-keun/opencv\\_tutorials.git](https://github.com/0-keun/opencv_tutorials.git)

## 2. Open CV 실습

### 1) 이미지 출력

- Python 환경에서 Open CV를 통해 다운받은 원본 이미지를 화면에 출력하기 (실습)
  - 주의 사항 1 : 이미지 파일의 확장자는 반드시 “jpg” 또는 “png” 파일을 사용할 것.
  - 주의 사항 2 : Python code 상에 작성하는 이미지의 경로를 정확하게 작성할 것.

```
1  import cv2
2
3  image = cv2.imread('./image/JackDaniels.jpg')
4
5  cv2.imshow("Bottle Image", image)
6  cv2.waitKey(0)
7  cv2.destroyAllWindows()
```

## 2. Open CV 실습

### 1) 이미지 출력

- Python 환경에서 Open CV를 통해 다운받은 원본 이미지를 화면에 출력하기 (실습)
  - 주의 사항 1 : 이미지 파일의 확장자는 반드시 “jpg” 또는 “png” 파일을 사용할 것.
  - 주의 사항 2 : Python code 상에 작성하는 이미지의 경로를 정확하게 작성할 것.



## 2. Open CV 실습

### 1) 이미지 출력

- 연습문제 1-1

- 이름 바꾸기, Image -> Bottle Image



## 2. Open CV 실습

### 1) 이미지 출력

- 연습문제 1-2
  - 이미지가 5초 후에 꺼지도록 하기
  - hint, `cv2.waitKey( )`에서 인수는 이미지가 켜져있는 시간[ms]을 의미하며 0은 이미지를 켜두는 것을 의미.

## 2. Open CV 실습

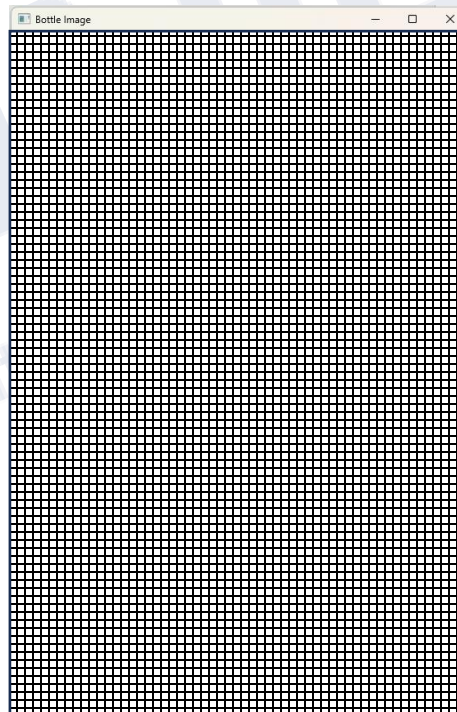
### 1) 이미지 출력

- 연습문제 1-3

- image 변수의 type과 변수 출력하기

```
print(type(image))  
print(image)
```

```
<class 'numpy.ndarray'>  
[[[143 165 176]  
  [143 165 176]  
  [143 165 176]  
  ...  
  [142 164 175]  
  [142 164 175]  
  [142 164 175]]  
  
[[[143 165 176]  
  [143 165 176]  
  [143 165 176]  
  ...  
  [143 165 176]  
  [143 165 176]  
  [143 165 176]]
```


$$\begin{bmatrix} [143,165,176] & \cdots & [142,164,175] \\ \vdots & \ddots & \vdots \\ [143,165,176] & \cdots & [143,165,176] \end{bmatrix}$$



## 2. Open CV 실습

### 2) 이미지 gray scale

- Python 환경에서 Open CV를 통해 원본 이미지를 gray scale화하여 화면에 출력하기 (실습)
  - 주의 사항 1 : 이미지 파일의 확장자는 반드시 “jpg” 또는 “png” 파일을 사용할 것.
  - 주의 사항 2 : Python code 상에 작성하는 이미지의 경로를 정확하게 작성할 것.

```
1  import cv2
2
3  image = cv2.imread('./image/JackDaniels.jpg')
4  gray_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
5
6  cv2.imshow("Image", image)
7  cv2.imshow("Gray_image", gray_image)
8  cv2.waitKey(0)
9  cv2.destroyAllWindows()
```

## 2. Open CV 실습

### 2) 이미지 gray scale

- Python 환경에서 Open CV를 통해 원본 이미지를 gray scale화하여 화면에 출력하기 (실습)
  - 주의 사항 1 : 이미지 파일의 확장자는 반드시 “jpg” 또는 “png” 파일을 사용할 것.
  - 주의 사항 2 : Python code 상에 작성하는 이미지의 경로를 정확하게 작성할 것.

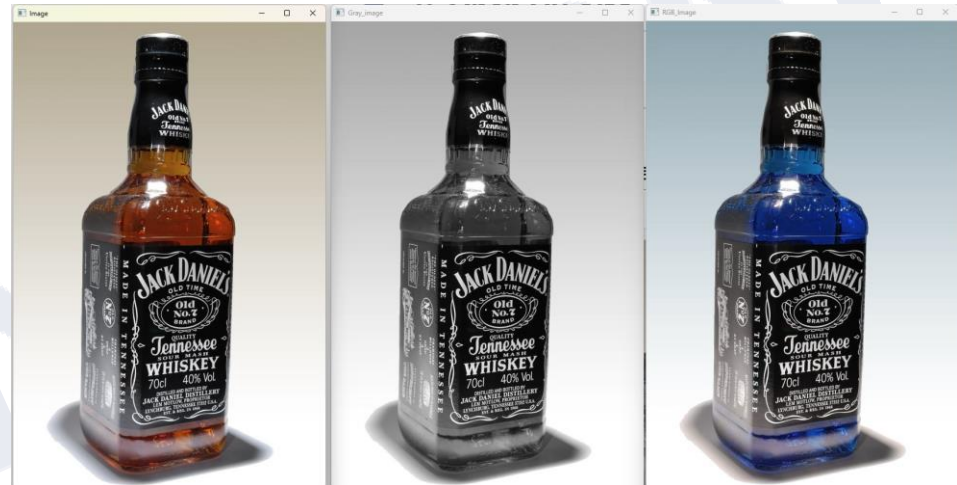


## 2. Open CV 실습

### 2) 이미지 gray scale

- 연습문제 2-1

- cv2.COLOR\_BGR2RGB로 색변환



- 연습문제 2-2

- (0,0)좌표의 데이터 비교, BGR\_image[0][0], RGB\_image[0][0], Gray\_image[0][0]을 비교해볼 것

```
BGR_image: [143 165 176]
RGB_image: [176 165 143]
Gray_image: 166
```

## 2. Open CV 실습

### 3) 이미지 edge 검출

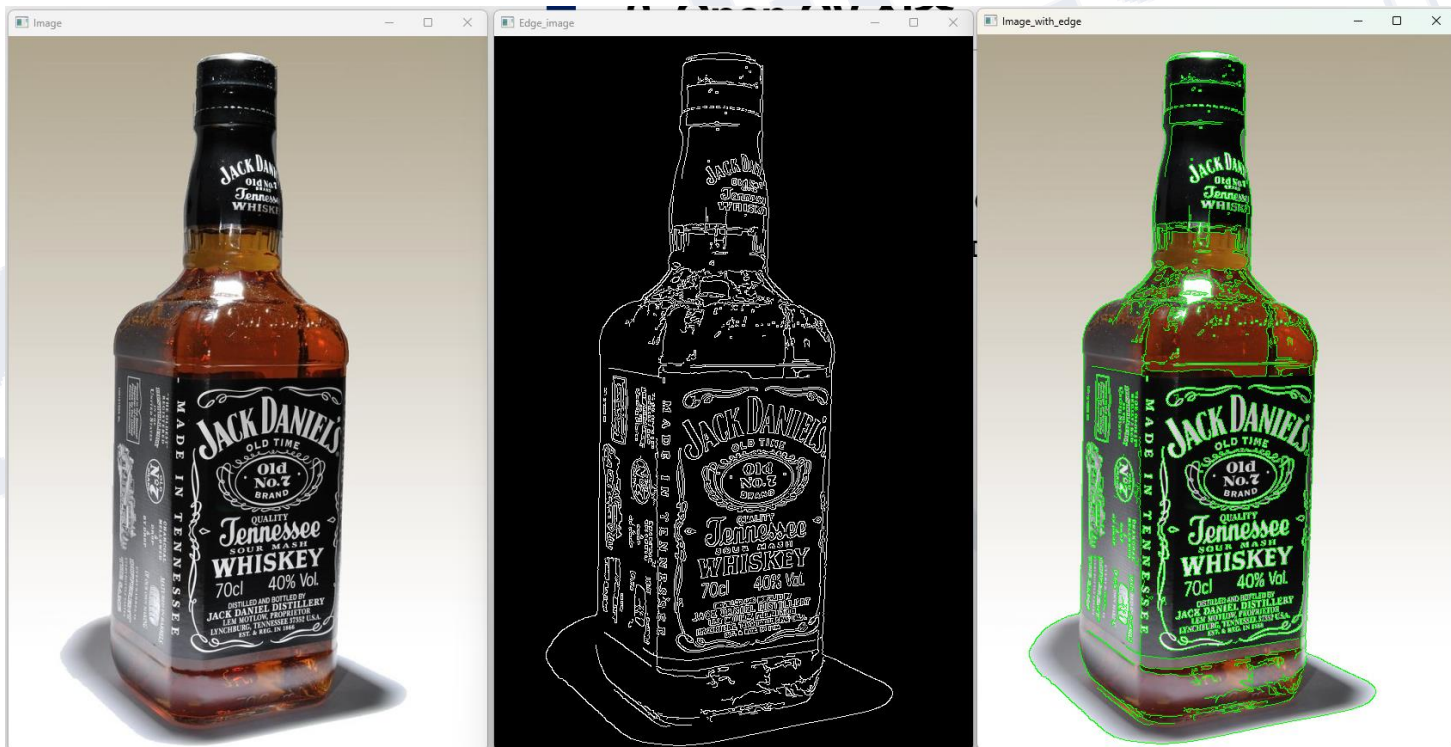
- Python 환경에서 Open CV를 통해 원본 이미지의 Edge를 검출하여 화면에 출력하기 (실습)
  - 주의 사항 1 : 이미지 파일의 확장자는 반드시 “jpg” 또는 “png” 파일을 사용할 것.
  - 주의 사항 2 : Python code 상에 작성하는 이미지의 경로를 정확하게 작성할 것.

```
1  import cv2
2
3  image = cv2.imread('./image/JackDaniels.jpg')
4  edge_image = cv2.Canny(image,50,300)
5  image_with_edge = image.copy()
6
7  contours, hierarchy = cv2.findContours(edge_image, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
8  cv2.drawContours(image_with_edge, contours, -1, (0,255,0), 1)
9
10 cv2.imshow("Image", image)
11 cv2.imshow("Edge_image", edge_image)
12 cv2.imshow("Image_with_edge", image_with_edge)
13 cv2.waitKey(0)
14 cv2.destroyAllWindows()
```

## 2. Open CV 실습

### 3) 이미지 edge 검출

- Python 환경에서 Open CV를 통해 원본 이미지의 Edge를 검출하여 화면에 출력하기 (실습)
  - 주의 사항 1 : 이미지 파일의 확장자는 반드시 “jpg” 또는 “png” 파일을 사용할 것.
  - 주의 사항 2 : Python code 상에 작성하는 이미지의 경로를 정확하게 작성할 것.



## 2. Open CV 실습

### 3) 이미지 edge 검출

- 연습문제 3-1

- threshold1, threshold2를 조절해보고 각 인수가 하는 역할 이해하기.

	threshold1	threshold2		threshold1	threshold2
Case 1	500	500	Case 4	300	300
Case 2	300	300	Case 5	150	300
Case 3	150	150	Case 6	50	300

## 2. Open CV 실습

### 4) 이미지 회전 / 크기 조절 / 기울이기

- Python 환경에서 Open CV를 원본 이미지를 변환(회전, 크기 조절, 기울이기) 후 출력하기 (실습)
  - 주의 사항 1 : 이미지 파일의 확장자는 반드시 “jpg” 또는 “png” 파일을 사용할 것.
  - 주의 사항 2 : Python code 상에 작성하는 이미지의 경로를 정확하게 작성할 것.

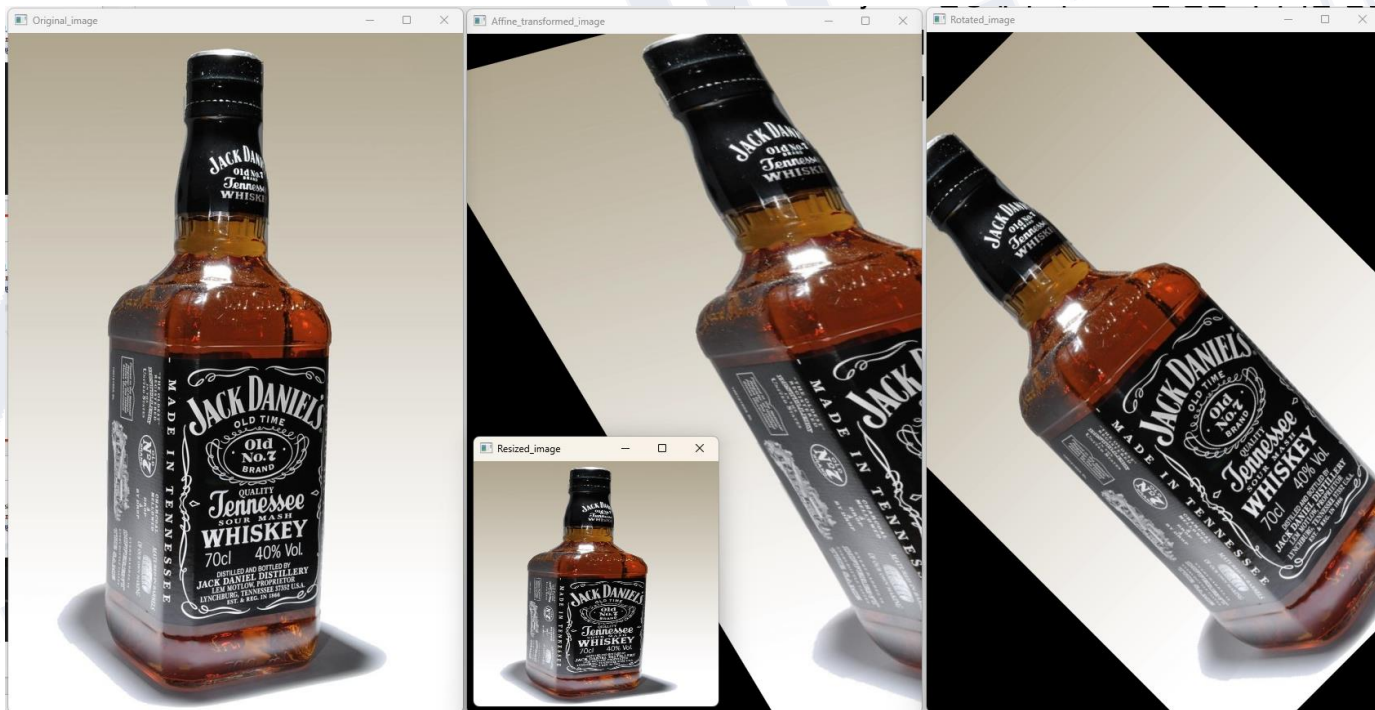
```
1  import cv2
2  import numpy as np
3
4  image = cv2.imread('./image/JackDaniels.jpg')
5  cv2.imshow('Original_image',image)
6
7  # Rotation
8  (h,w) = image.shape[:2]
9  center = (w/2, h/2)
10 M = cv2.getRotationMatrix2D(center,45,1.0)
11 rotated_image = cv2.warpAffine(image,M,(w,h))
12 cv2.imshow('Rotated_image',rotated_image)
13
14 # Resize
15 size = 300
16 resized_image = cv2.resize(image,(size,size))
17 cv2.imshow('Resized_image',resized_image)
18
19 # Affine Transform
20 pts1 = np.float32([[50,50],[200,50],[50,200]])
21 pts2 = np.float32([[10,100],[200,50],[100,250]])
22 M = cv2.getAffineTransform(pts1,pts2)
23 affine_image = cv2.warpAffine(image,M,(w,h))
24 cv2.imshow('Affine_transformed_image',affine_image)
25
26 cv2.waitKey(0)
27 cv2.destroyAllWindows()
```



## 2. Open CV 실습

### 4) 이미지 회전 / 크기 조절 / 기울이기

- Python 환경에서 Open CV를 원본 이미지를 변환(회전, 크기 조절, 기울이기) 후 출력하기 (실습)
  - 주의 사항 1 : 이미지 파일의 확장자는 반드시 “jpg” 또는 “png” 파일을 사용할 것.
  - 주의 사항 2 : Python code 상에 작성하는 이미지의 경로를 정확하게 작성할 것.





## 2. Open CV 실습

### 4) 이미지 회전 / 크기 조절 / 기울이기

- 연습문제 4-1

- resize 사이즈를 2배 키워서 출력해보기

- 1:2 비율로 출력해보기



## 2. Open CV 실습

### 5) 이미지 filtering

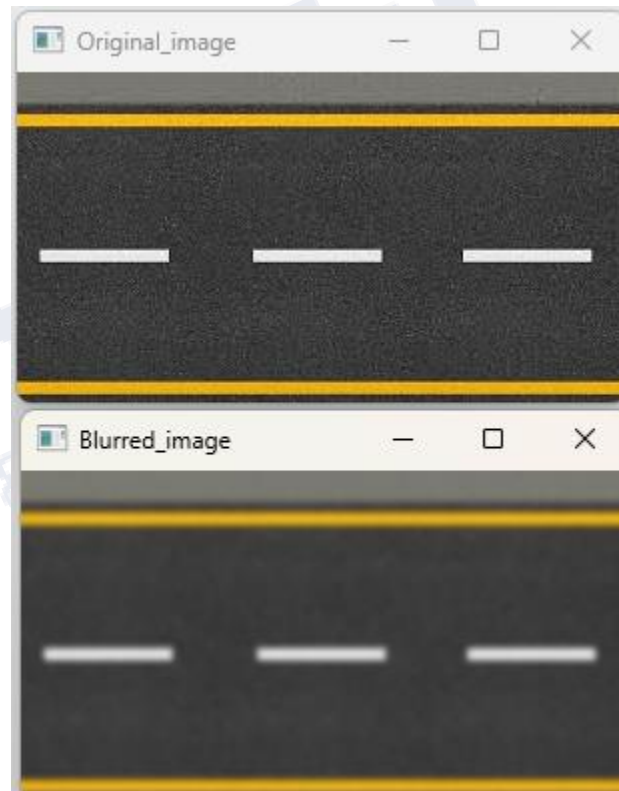
- Python 환경에서 Open CV를 통해 원본 이미지를 Filtering(Gaussian blur)하여 출력하기 (실습)
  - 주의 사항 1 : 이미지 파일의 확장자는 반드시 “jpg” 또는 “png” 파일을 사용할 것.
  - 주의 사항 2 : Python code 상에 작성하는 이미지의 경로를 정확하게 작성할 것.

```
1  import cv2
2
3  image = cv2.imread('./image/road.jpg')
4  cv2.imshow('Original_image',image)
5
6  blurred_image = cv2.GaussianBlur(image,(5,5),0)
7  cv2.imshow('Blurred_image',blurred_image)
8
9  cv2.waitKey(0)
10 cv2.destroyAllWindows()
```

## 2. Open CV 실습

### 5) 이미지 filtering

- Python 환경에서 Open CV를 통해 원본 이미지를 Filtering(Gaussian blur)하여 출력하기 (실습)
  - 주의 사항 1 : 이미지 파일의 확장자는 반드시 “jpg” 또는 “png” 파일을 사용할 것.
  - 주의 사항 2 : Python code 상에 작성하는 이미지의 경로를 정확하게 작성할 것.



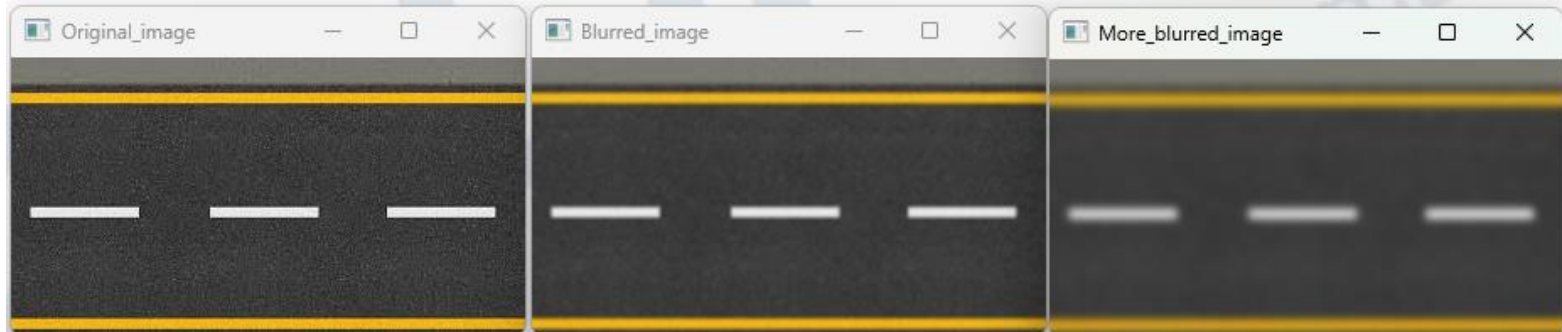
## 2. Open CV 실습

### 5) 이미지 filtering

- 연습문제 5-1

- 이미지를 더 흐릿하게 조정하기

```
6 blurred_image = cv2.GaussianBlur(image, (5,5), 0)
```



## 2. Open CV 실습

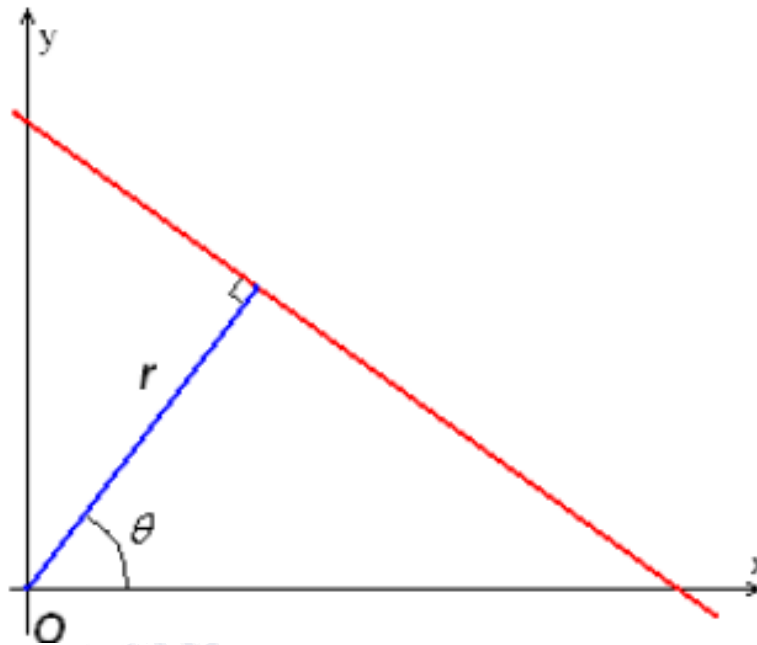
### 6) 차선 인식 ( Hough / HSV / Hough + HSV )



- 위 도로 이미지 속 차선을 Hough transform, HSV(Hue + Saturation + Value) 색 공간 등의 Open CV 기반의 기법을 사용하여 검출

## 2. Open CV 실습

### 6) 차선 인식 (Hough / HSV / Hough + HSV)



- Hough transform 기법은 Computer Vision과 Image Process에서 선, 원, 사각형 등의 도형을 검출하는 데 사용되는 기법
- 직선 검출 과정  
: 이미지 gray scale → 이미지 필터링(Gaussian Blur) → 이미지 Edge 검출 → Hough transform

## 2. Open CV 실습

### 6) 차선 인식 (Hough / HSV / Hough + HSV)

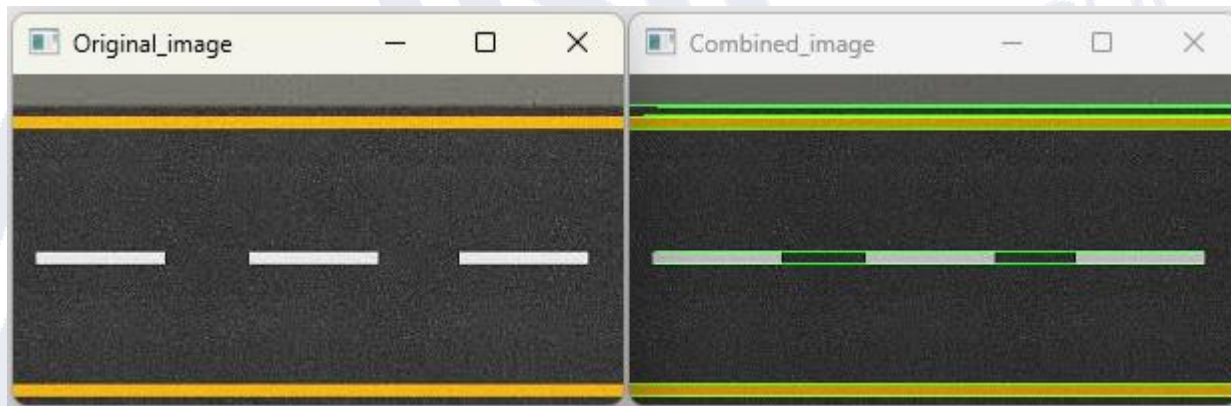
- Python 환경에서 Open CV의 내장 함수를 이용하여 도로 이미지 속 차선 검출 수행하기 (실습)
  - 주의 사항 1 : Gray scale, Gaussian blur(filtering), Edge detection, Hough transform 사용
  - 주의 사항 2 : 도로 이미지에 따라 검출 결과의 정확도가 다를 수 있음.

```
1  import cv2
2  import numpy as np
3
4  image_path = './image/road.jpg'
5  image = cv2.imread(image_path)
6
7  gray_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
8  blurred_image = cv2.GaussianBlur(gray_image, (5,5), 0)
9  edge_image = cv2.Canny(blurred_image, 50, 150)
10
11 lines = cv2.HoughLinesP(edge_image, 1, np.pi/180, 20, minLineLength=30, maxLineGap=200)
12 line_image = np.zeros_like(image)
13
14
15 if lines is not None:
16     for line in lines:
17         x1, y1, x2, y2 = line[0]
18         cv2.line(line_image, (x1,y1), (x2,y2), (0,255,0), 1)
19
20     combined_image = cv2.addWeighted(image, 0.8, line_image, 1, 1)
21     # _combined_image = cv2.add(image, line_image)
22     cv2.imshow('Original_image', image)
23     cv2.imshow('Combined_image', combined_image)
24     # cv2.imshow('_Combined_image', _combined_image)
25     cv2.waitKey(0)
26     cv2.destroyAllWindows()
27 else:
28     print('There is no lane.')
```

## 2. Open CV 실습

### 6) 차선 인식 (Hough / HSV / Hough + HSV)

- Python 환경에서 Open CV의 내장 함수를 이용하여 도로 이미지 속 차선 검출 수행하기 (실습)
  - 주의 사항 1 : Gray scale, Gaussian blur(filtering), Edge detection, Hough transform 사용
  - 주의 사항 2 : 도로 이미지에 따라 검출 결과의 정확도가 다를 수 있음.





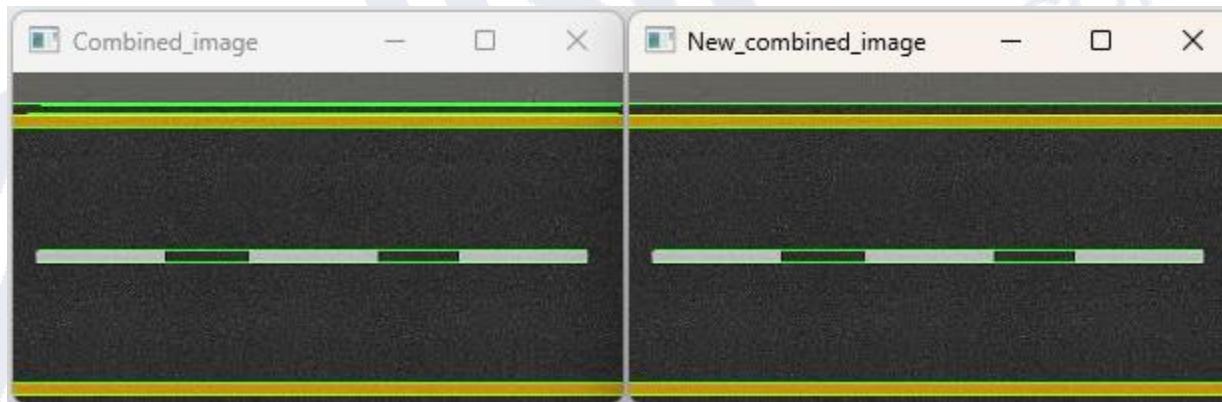
## 2. Open CV 실습

### 6) 차선 인식 (Hough / HSV / Hough + HSV)

- 연습문제 6-1

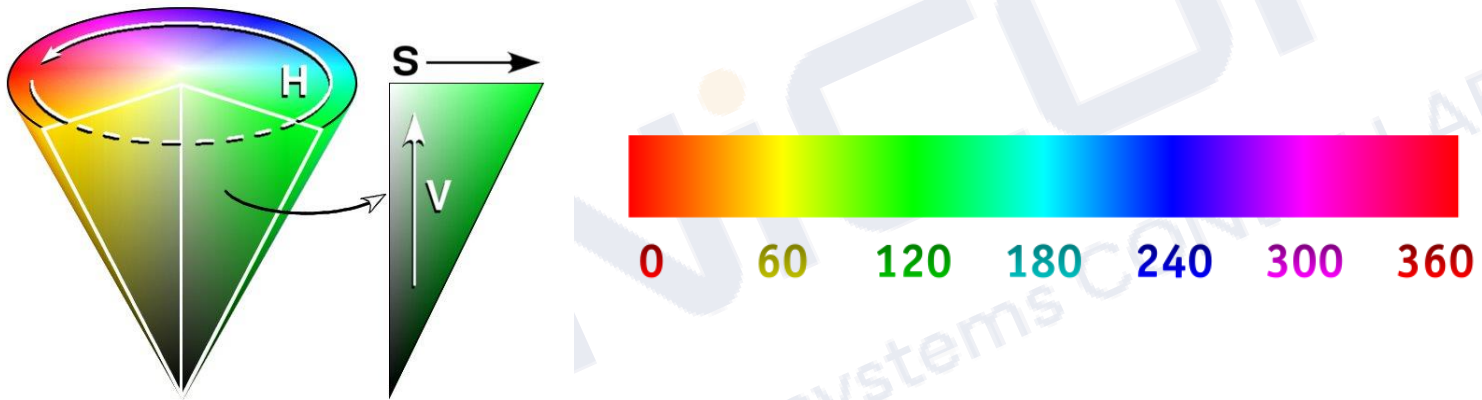
- HoughLinesP() 함수 threshold 값 조정하기

```
lines = cv2.HoughLinesP(edge_image,1,np.pi/180,150,minLineLength=30,maxLineGap=200)
```



## 2. Open CV 실습

### 6) 차선 인식 [ Hough / HSV / Hough + HSV ]



- HSV(Hue + Saturation + Value) 모델은 인간의 색 인지에 기반을 둔 색상 모델
- HSV는 Hue(색상), Saturation(채도), Value(명도)를 기반으로 하는 색 공간
- 특정 색상 범위를 쉽게 지정하여 필터링(filtering)할 수 있다는 이점 존재

## 2. Open CV 실습

### 6) 차선 인식 [ Hough / HSV / Hough + HSV ]

- Python 환경에서 Open CV의 내장 함수를 이용하여 도로 이미지 속 차선 검출 수행하기 (실습)
  - 주의 사항 1 : HSV 색상 공간을 사용하여 도로 이미지 속 차선(노란색 차선, 흰색 차선)을 검출할 것.
  - 주의 사항 2 : 도로 이미지에 따라 검출 결과의 정확도가 다를 수 있음.

```
1 import cv2
2 import numpy as np
3
4 image_path = './image/road.jpg'
5 original_image = cv2.imread(image_path)
6 hsv_image = cv2.cvtColor(original_image, cv2.COLOR_BGR2HSV)
7
8 yellow_lower = np.array([20, 100, 100])
9 yellow_upper = np.array([30, 255, 255])
10
11 white_lower = np.array([0, 0, 200])
12 white_upper = np.array([180, 25, 255])
13
14 def yellow_filter(image):
15     yellow_mask = cv2.inRange(hsv_image, yellow_lower, yellow_upper)
16     yellow_image = cv2.bitwise_and(image, image, mask=yellow_mask)
17     cv2.imshow('yellow_image', yellow_image)
18     return yellow_image
19
20 def white_filter(image):
21     white_mask = cv2.inRange(hsv_image, white_lower, white_upper)
22     white_image = cv2.bitwise_and(image, image, mask=white_mask)
23     cv2.imshow('white_image', white_image)
24     return white_image
25
26 def combine_filter(image):
27     yellow_mask = cv2.inRange(hsv_image, yellow_lower, yellow_upper)
28     white_mask = cv2.inRange(hsv_image, white_lower, white_upper)
29     combined_mask = cv2.bitwise_or(yellow_mask, white_mask)
30     combined_image = cv2.bitwise_and(image, image, mask=combined_mask)
31     cv2.imshow('combined_image', combined_image)
32     return combined_image
33
34 yellow_image = yellow_filter(original_image)
35 white_image = white_filter(original_image)
36 combined_image = combine_filter(original_image)
37
38 cv2.imshow('original_image', original_image)
39 cv2.waitKey(0)
40 cv2.destroyAllWindows()
```

## 2. Open CV 실습

### 6) 차선 인식 [ Hough / HSV / Hough + HSV ]

- Python 환경에서 Open CV의 내장 함수를 이용하여 도로 이미지 속 차선 검출 수행하기 (실습)
  - 주의 사항 1 : HSV 색상 공간을 사용하여 도로 이미지 속 차선(노란색 차선, 흰색 차선)을 검출할 것.
  - 주의 사항 2 : 도로 이미지에 따라 검출 결과의 정확도가 다를 수 있음.



## 2. Open CV 실습

### 6) 차선 인식 [ Hough / HSV / Hough + HSV ]

- 연습문제 7-1

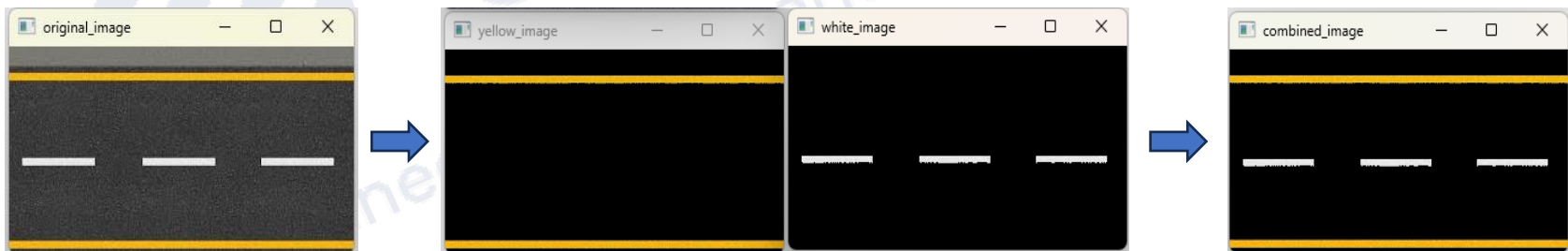
- white mask 생성 후 합치기

hint.

- HSV\_scroll을 이용해서 white\_lower, white\_upper 구하기

- white mask 생성, 이미지 합치기

```
combined_mask = cv2.bitwise_or(yellow_mask, white_mask)  
combined_image = cv2.bitwise_and(image, image, mask=combined_mask)
```



## 2. Open CV 실습

### 6) 차선 인식 [ Hough / HSV / **Hough + HSV** ]

Method	Characteristic
Hough	<ul style="list-style-type: none"><li>▪ Gray scale로 인한 색상 정보 손실</li><li>▪ Edge 검출 수행 시, noise에 민감</li></ul>
HSV	<ul style="list-style-type: none"><li>▪ 조명 변화에 민감</li><li>▪ 차선 검출 시, HSV 범위 설정에 민감</li></ul>
Hough + HSV	<ul style="list-style-type: none"><li>▪ Noise 및 조명 변화에 대한 강건성 증가</li><li>▪ 다른 객체와의 혼동 감소</li></ul>

## 2. Open CV 실습

### 6) 차선 인식 [ Hough / HSV / Hough + HSV ]

- Python 환경에서 Open CV의 내장 함수를 이용하여 도로 이미지 속 차선 검출 수행하기 (실습)
  - 주의 사항 1 : Hough transform과 HSV 색상 공간을 모두 사용하여 이미지 속 차선을 검출할 것.
  - 주의 사항 2 : 도로 이미지에 따라 검출 결과의 정확도가 다를 수 있음.

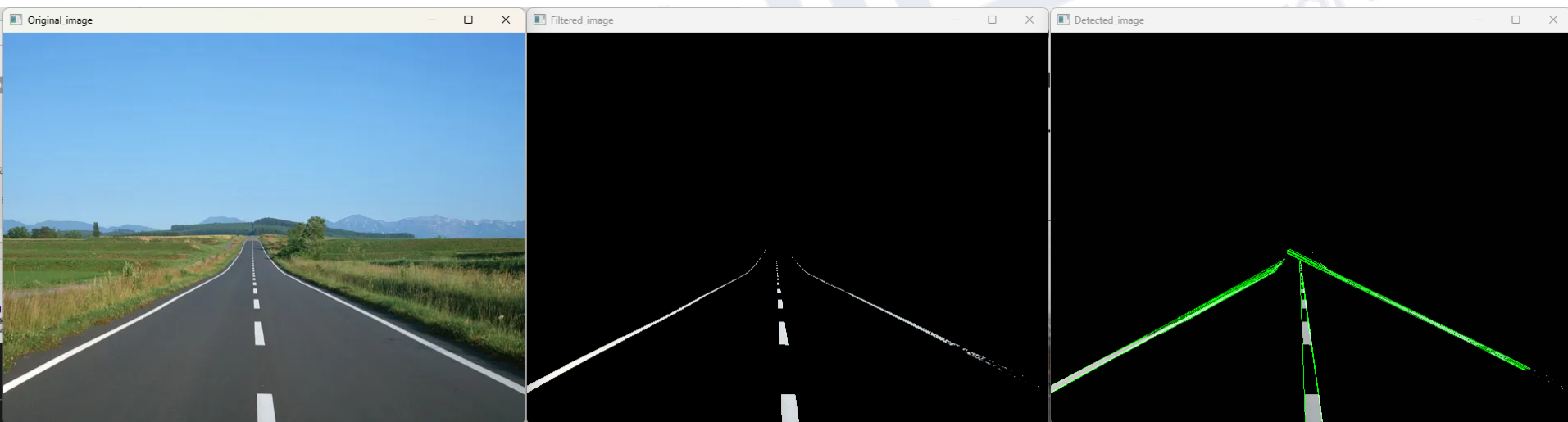
```

1  import cv2
2  import numpy as np
3
4  ## HSV
5  def HSV_image(image):
6      hsv_image = cv2.cvtColor(image, cv2.COLOR_BGR2HSV)
7
8      white_lower = np.array([0, 0, 200])
9      white_upper = np.array([180, 255, 255])
10     white_mask = cv2.inRange(hsv_image, white_lower, white_upper)
11
12     filtered_image = cv2.bitwise_and(image, image, mask=white_mask)
13
14     return filtered_image
15
16  ## Hough
17  def lane_detect(image):
18      gray_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
19      blurred_image = cv2.GaussianBlur(gray_image, (5, 5), 0)
20      edge_image = cv2.Canny(blurred_image, 50, 150)
21
22      lines = cv2.HoughLinesP(edge_image, 1, np.pi/180, 20, minLineLength=30, maxLineGap=200)
23      line_image = np.zeros_like(image)
24
25      if lines is not None:
26          for line in lines:
27              x1, y1, x2, y2 = line[0]
28              cv2.line(line_image, (x1, y1), (x2, y2), (0, 255, 0), 1)
29
30              combined_image = cv2.addWeighted(image, 0.8, line_image, 1, 1)
31
32      else:
33          combined_image = image
34
35      return combined_image
36
37  image_path = './image/road_camera.jpg'
38  image = cv2.imread(image_path)
39
40  filtered_image = HSV_image(image)
41  detected_image = lane_detect(filtered_image)
42
43  cv2.imshow('Original_image', image)
44  cv2.imshow('Filtered_image', filtered_image)
45  cv2.imshow('Detected_image', detected_image)
46  cv2.waitKey(0)
47  cv2.destroyAllWindows()
  
```

## 2. Open CV 실습

### 6) 차선 인식 [ Hough / HSV / Hough + HSV ]

- Python 환경에서 Open CV의 내장 함수를 이용하여 도로 이미지 속 차선 검출 수행하기 (실습)
  - 주의 사항 1 : Hough transform과 HSV 색상 공간을 모두 사용하여 이미지 속 차선을 검출할 것.
  - 주의 사항 2 : 도로 이미지에 따라 검출 결과의 정확도가 다를 수 있음.



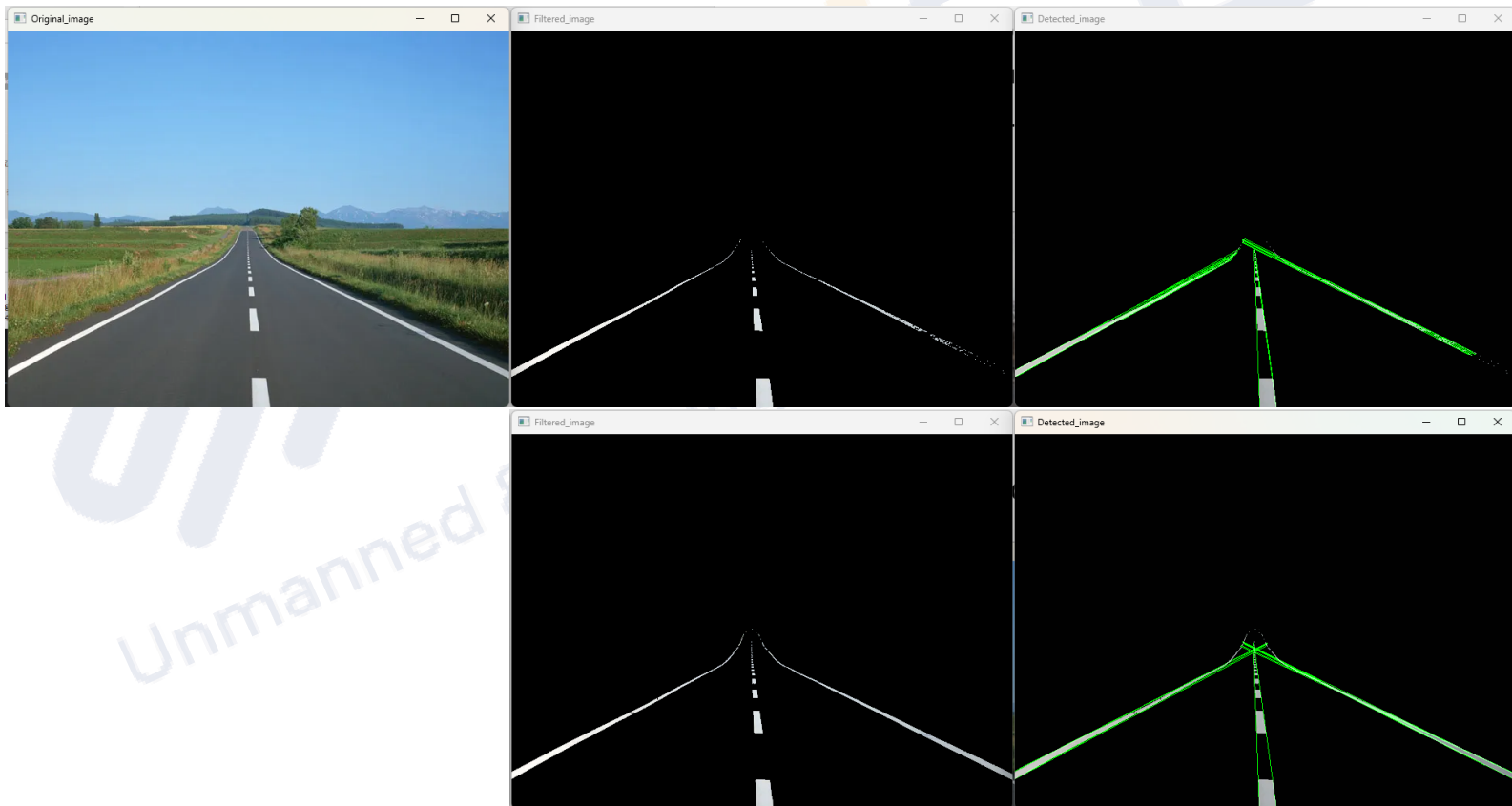


## 2. Open CV 실습

### 6) 차선 인식 [ Hough / HSV / Hough + HSV ]

- 연습문제 8-1

- 지금까지 배운 내용을 이용해서 차선 검출 뚜렷하게 하기 (색 조정, edge, lines 파라미터 조정 등)

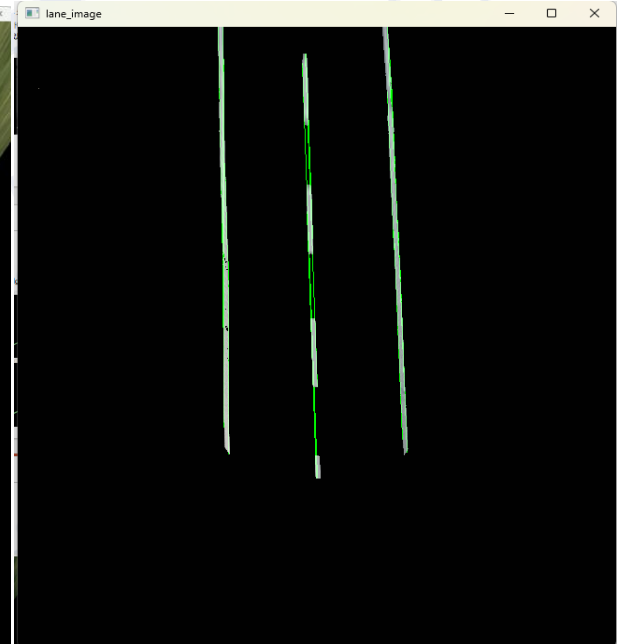
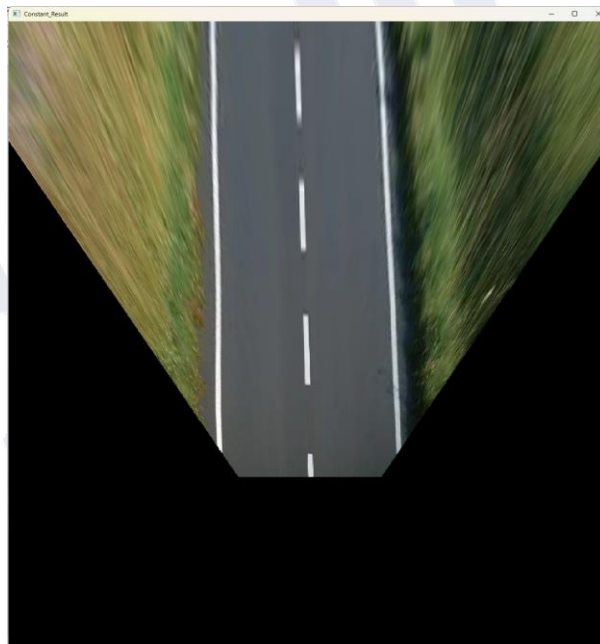
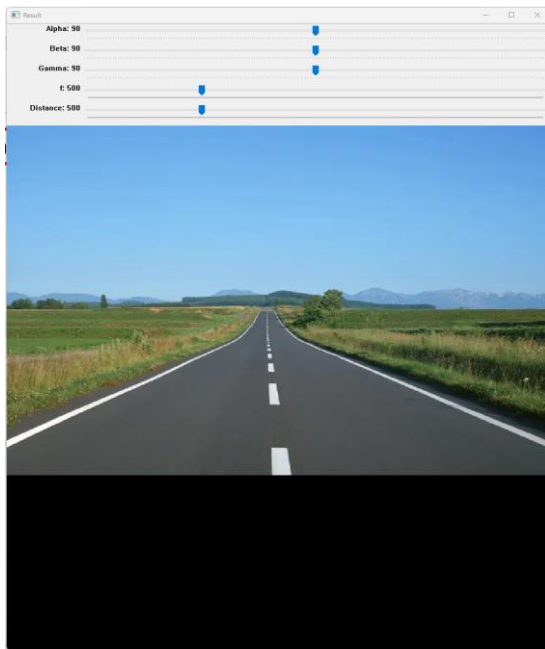


## 2. Open CV 실습

### 6) 차선 인식 ( Hough / HSV / **Hough + HSV** )

- 연습문제 8-2

- Bird Eye View와 합쳐보기



## 2. Open CV 실습

### 7) LIMO 카메라 사용

- Python 환경에서 Open CV의 내장 함수를 이용하여 카메라 영상 속 차선 검출 수행하기 (실습)
  - 주의 사항 1 : Gray scale, Gaussian blur(filtering), Edge detection, Hough transform 사용
  - 주의 사항 2 : 도로 이미지에 따라 검출 결과의 정확도가 다를 수 있음.

# QnA

UNMILCON  
Unmanned & Intelligent systems CONTROL LAB

**Thank you.**

