

Take Out 5주차 OpenCv

제작_26기 송우석

목차

01 OpenCV

03 Filtering

05 with Camera

02 BEV

04 Edge & Line

01 OpenCV OpenCV란?

- Open Computer Vision library의 약자로, 실시간 영상 및 이미지 처리를 위한 오픈 소스 라이브러리를 의미
- 영상처리, 머신러닝, OCR(광학 문자 인식), 기하(Geometry) 변환 등의 다양한 라이브러리들이 포함되어 있음
- 딥러닝 기반의 얼굴인식, 물체 인식 등의 라이브러리도 지원



Image
Processing



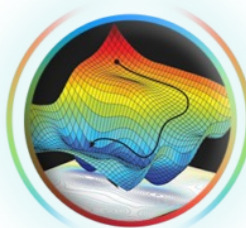
Feature Detection
& Matching



Object
Detection



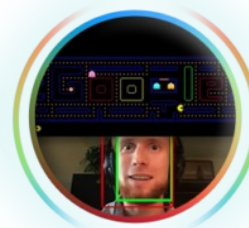
Geometry



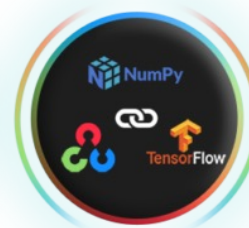
Machine
Learning



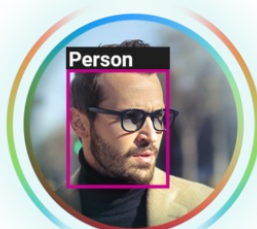
Video
Analysis



GUI Tools



Integration



Facial Recognition



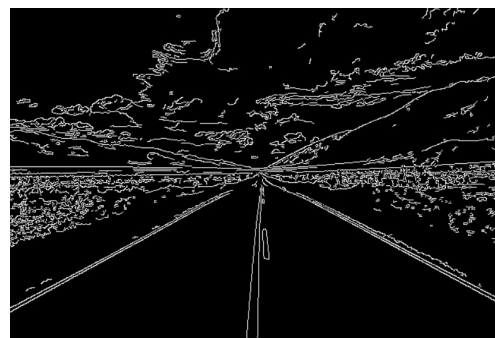
Object Detection



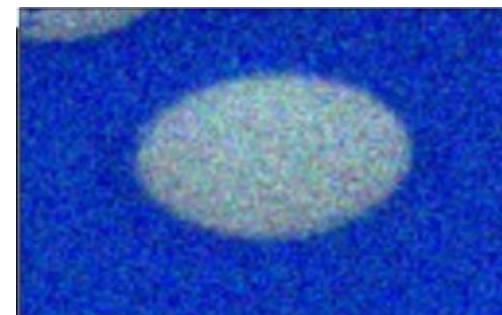
Augmented Reality

01 OpenCV 사용하는 이유

- 보통 이미지 데이터를 다룬다고 한다면, '픽셀' 단위로 다룸
- OpenCV는 이러한 연속적인 픽셀들을 **numpy 행렬**로 다룸으로서, 연산/필터링/좌표 변환 등의 task들을 빠르게 수행할 수 있게 해 줌
- 또한 AI 모델 없이도 물체 인식과 같은 task를 색상, 형태, 에지 등의 **Rule-Based 기반**으로 해결 할 수 있게 해 줌



edge detection



denoising

01 OpenCV Opencv 설치

<ROS 2 빌드관리 툴 및 Python 패키지 툴 설치>

update

\$ sudo apt update

패키지 설치

```
$ sudo apt install -y \  
python3-colcon-common-extensions \  
python3-rosdep \  
python3-vcstool \  
python3-pip
```

```
wooseok@wooseok-17ZD90Q-GX5LK:~$ sudo apt install -y python3-colcon-common-extensions \  
> python3-rosdep \  
> python3-vcstool \  
> python3-pip  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
python3-pip is already the newest version (22.0.2+dfsg-1ubuntu0.7).  
python3-colcon-common-extensions is already the newest version (0.3.0-100).  
python3-rosdep is already the newest version (0.26.0-1).  
python3-vcstool is already the newest version (0.3.0-1).  
The following packages were automatically installed and are no longer required:  
  libode-dev linux-headers-6.8.0-64-generic linux-hwe-6.8-headers-6.8.0-64 linux-hwe-6.8-tools-6.8.0-64  
  linux-image-6.8.0-64-generic linux-modules-6.8.0-64-generic linux-modules-extra-6.8.0-64-generic  
  linux-tools-6.8.0-64-generic  
Use 'sudo apt autoremove' to remove them.  
0 upgraded, 0 newly installed, 0 to remove and 517 not upgraded.  
wooseok@wooseok-17ZD90Q-GX5LK:~$
```

01 OpenCV Opencv 설치

<관련 의존성 패키지 설치>

패키지 설치

```
$ sudo apt install -y \  
ros-humble-rclpy \  
ros-humble-sensor-msgs \  
ros-humble-cv-bridge
```

```
wooseok@wooseok-17ZD90Q-GX5LK:~$ sudo apt install -y ros-humble-rclpy ros-humble-sensor-msgs ros-humble-cv-bridge  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
The following packages were automatically installed and are no longer required:  
  libode-dev linux-headers-6.8.0-64-generic linux-hwe-6.8-headers-6.8.0-64 linux-hwe-6.8-tools-6.8.0-64  
  linux-image-6.8.0-64-generic linux-modules-6.8.0-64-generic linux-modules-extra-6.8.0-64-generic  
  linux-tools-6.8.0-64-generic  
Use 'sudo apt autoremove' to remove them.  
The following packages will be upgraded:  
  ros-humble-cv-bridge ros-humble-rclpy ros-humble-sensor-msgs  
3 upgraded, 0 newly installed, 0 to remove and 514 not upgraded.  
Need to get 773 kB of archives.  
After this operation, 1,024 B of additional disk space will be used.  
Get:1 http://packages.ros.org/ros2/ubuntu jammy/main amd64 ros-humble-sensor-msgs amd64 4.9.0-1jammy.20250823.083704 [305 kB]  
Get:2 http://packages.ros.org/ros2/ubuntu jammy/main amd64 ros-humble-cv-bridge amd64 3.2.1-1jammy.20250823.090127 [110 kB]  
Get:3 http://packages.ros.org/ros2/ubuntu jammy/main amd64 ros-humble-rclpy amd64 3.3.17-1jammy.20250823.093318 [359 kB]  
Fetched 773 kB in 7s (105 kB/s)  
(Reading database ... 372719 files and directories currently installed.)  
Preparing to unpack .../ros-humble-sensor-msgs_4.9.0-1jammy.20250823.083704_amd64.deb ...  
Unpacking ros-humble-sensor-msgs (4.9.0-1jammy.20250823.083704) over (4.9.0-1jammy.20250617.204740) ...  
Preparing to unpack .../ros-humble-cv-bridge_3.2.1-1jammy.20250823.090127_amd64.deb ...  
Unpacking ros-humble-cv-bridge (3.2.1-1jammy.20250823.090127) over (3.2.1-1jammy.20250617.212054) ...  
Preparing to unpack .../ros-humble-rclpy_3.3.17-1jammy.20250823.093318_amd64.deb ...  
Unpacking ros-humble-rclpy (3.3.17-1jammy.20250823.093318) over (3.3.16-1jammy.20250607.033641) ...  
Setting up ros-humble-sensor-msgs (4.9.0-1jammy.20250823.083704) ...  
Setting up ros-humble-rclpy (3.3.17-1jammy.20250823.093318) ...  
Setting up ros-humble-cv-bridge (3.2.1-1jammy.20250823.090127) ...  
Processing triggers for libc-bin (2.35-0ubuntu3.10) ...
```

01 OpenCV Opencv 설치

<Python OpenCV 바인딩 설치>

패키지 설치

```
$ sudo apt install python3-opencv
```

패키지 확인

```
$ python3
```

```
>>import cv2
```

```
>>cv2.__version__
```

```
wooseok@wooseok-17ZD90Q-GX5LK:~$ sudo apt install python3-opencv
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
python3-opencv is already the newest version (4.5.4+dfsg-9ubuntu4).
The following packages were automatically installed and are no longer required:
  libode-dev linux-headers-6.8.0-64-generic linux-hwe-6.8-headers-6.8.0-64 linux-hwe-6.8-tools-6.8.0-64 linux-image-6.8.0-64-generic
  linux-modules-6.8.0-64-generic linux-modules-extra-6.8.0-64-generic linux-tools-6.8.0-64-generic
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 514 not upgraded.
wooseok@wooseok-17ZD90Q-GX5LK:~$ python3
Python 3.10.12 (main, Aug 15 2025, 14:32:43) [GCC 11.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import cv2
>>> cv2.__version__
'4.12.0-dev'
>>>
```

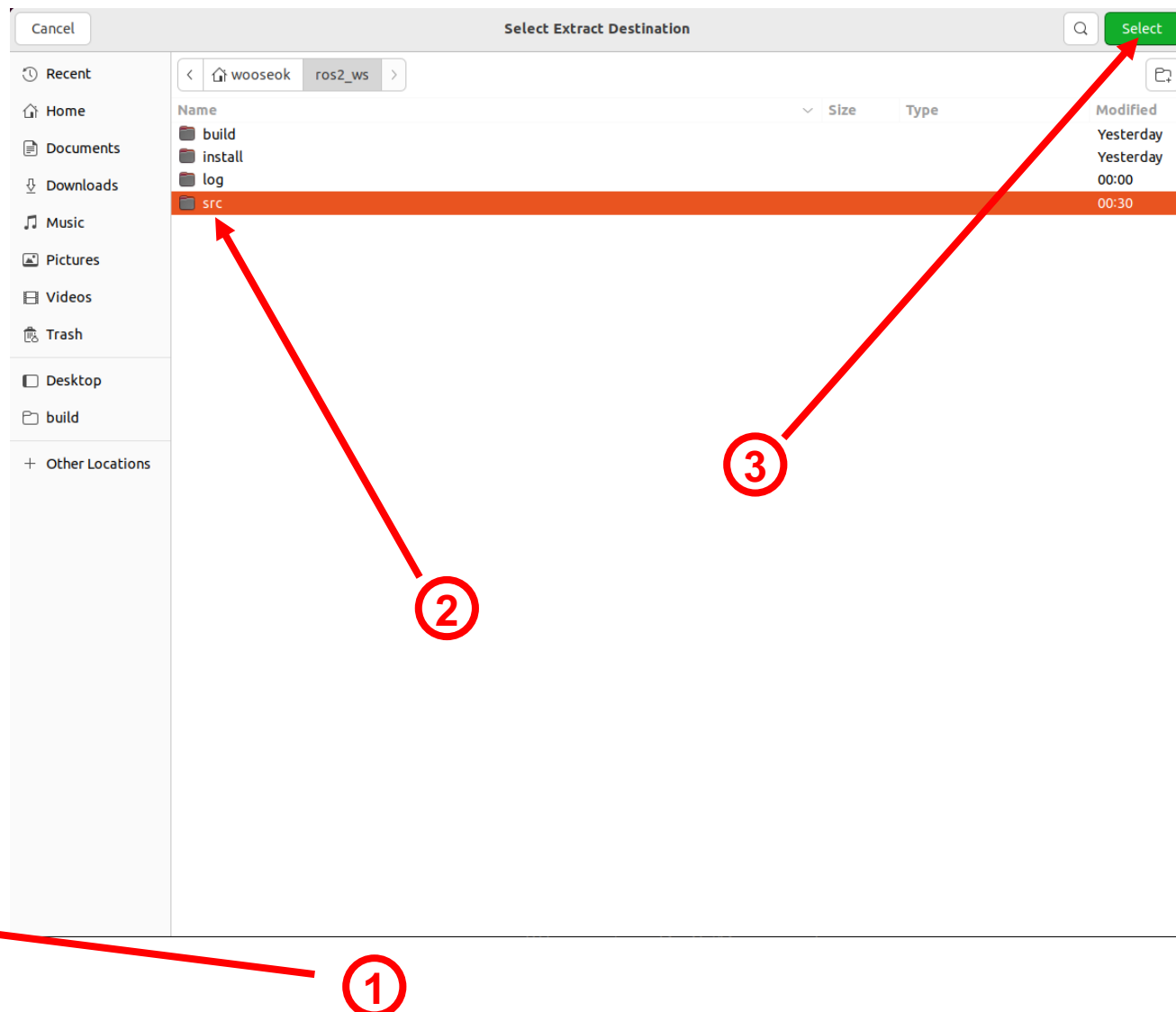
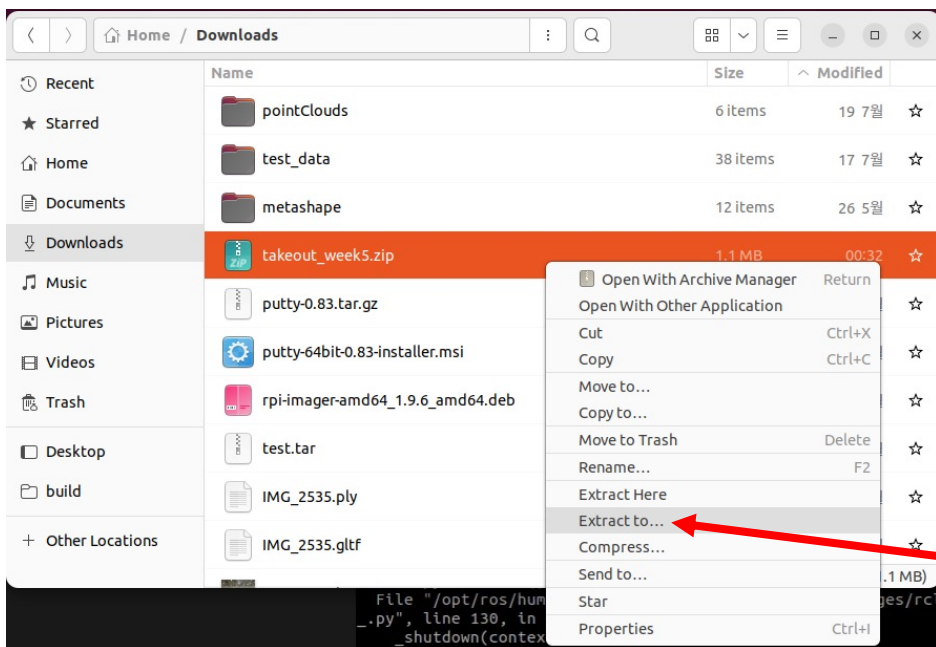
01 Open CV 필수 패키지 설치

<takeout_week5 패키지 다운 & 경로설정>

패키지 설치

```
$ cd ~/ros2_ws/src
```

이 경로에 압축 풀기



01 Open CV 필수 패키지 설치

<takeout_week5 패키지 빌드>

패키지 빌드

```
$ cd ~/ros2_ws  
$ colcon build --packages-select takeout_week5  
$ source install/setup.bash
```

패키지 확인

```
$ ros2 pkg list | grep takeout_week5
```

```
wooseok@wooseok-17ZD90Q-GX5LK:~/ros2_ws$ colcon build --packages-select takeout_week5  
Starting >>> takeout_week5  
Finished <<< takeout_week5 [0.77s]  
  
Summary: 1 package finished [1.01s]  
wooseok@wooseok-17ZD90Q-GX5LK:~/ros2_ws$ source install/setup.bash
```

```
wooseok@wooseok-17ZD90Q-GX5LK:~/ros2_ws$ ros2 pkg list | grep takeout_we  
ek5  
takeout_week5  
wooseok@wooseok-17ZD90Q-GX5LK:~/ros2_ws$
```

패키지 명이 뜨면 성공!

02 BEV BEV란?



BEV

- Bird Eye View의 약자로, 원근 변환을 통해 물체를 위에서 바라보는 듯한 효과를 만들어줌
- opencv에서는 `cv2.getPerspectiveTransform()`와 `cv2.warpPerspective()`함수를 통해 구현 가능
- 이미지상에서 원하는 연산 영역을 지정하여 적용 → ROI 설정

02 BEV BEV 실습

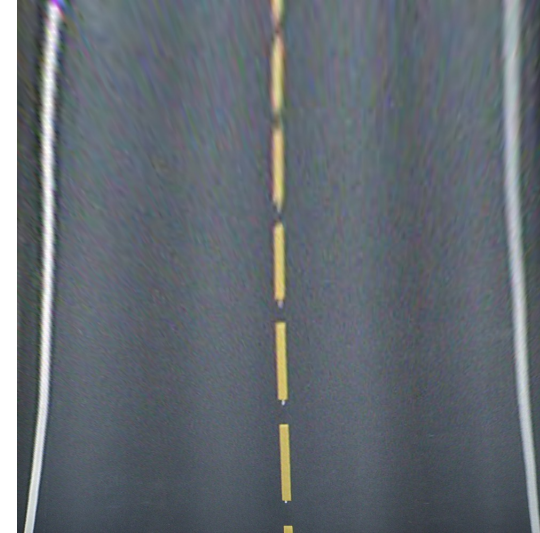
bev 코드 실행

```
$ ros2 run takeout_week5 bev
```

```
wooseok@wooseok-17ZD90Q-GX5LK:~/ros2_ws$ ros2 run takeout_week5 bev
```



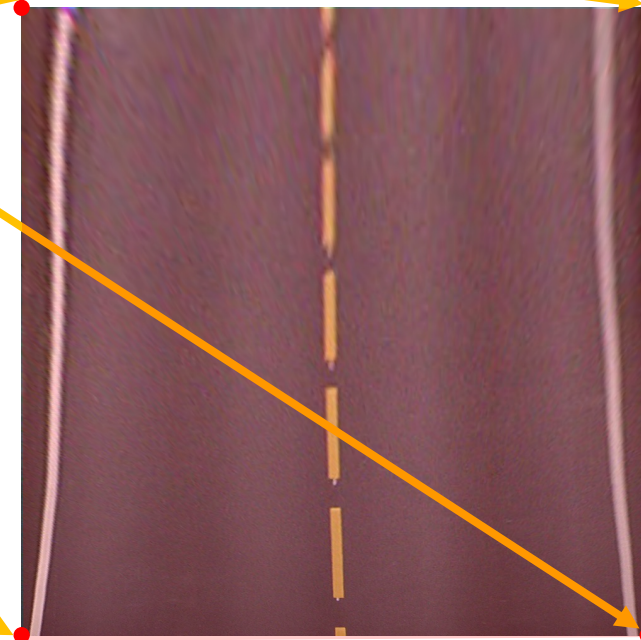
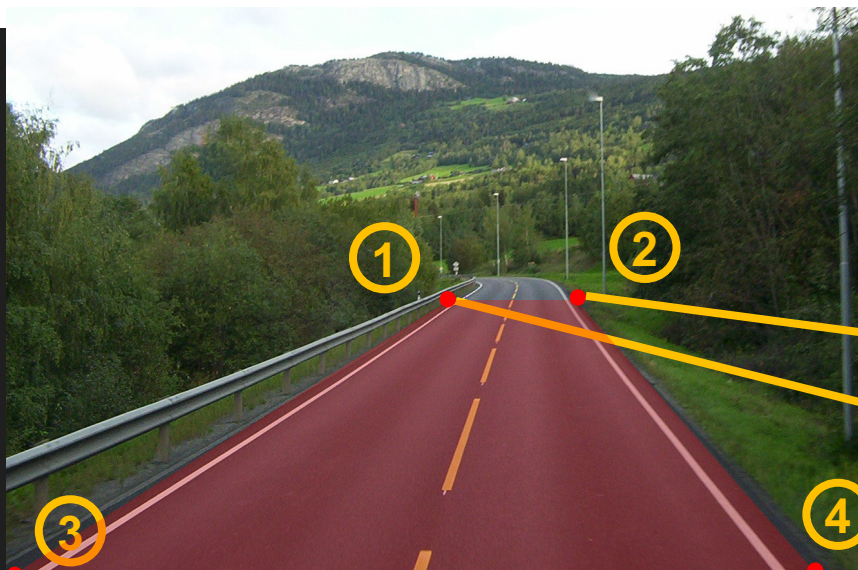
Original Image



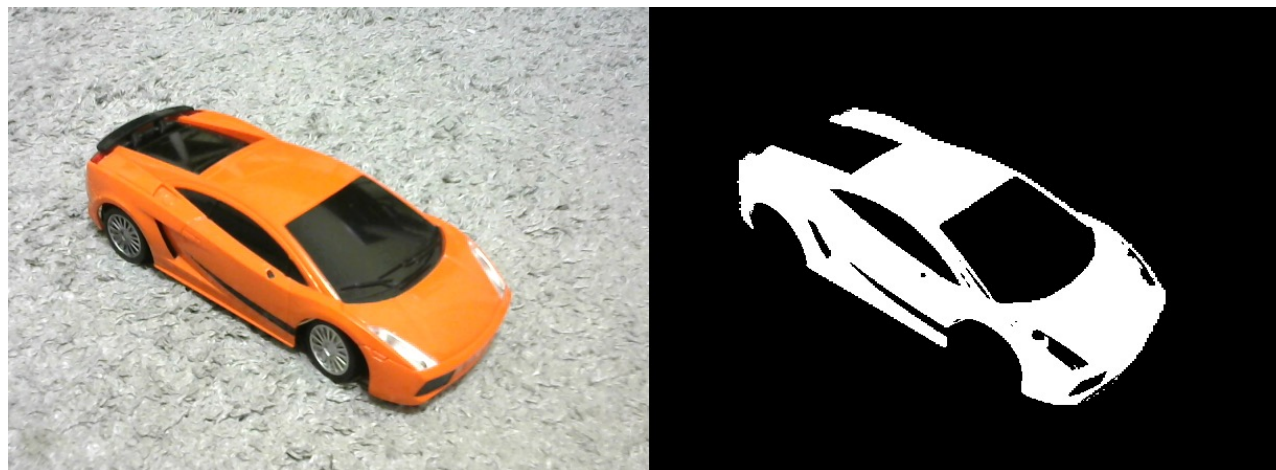
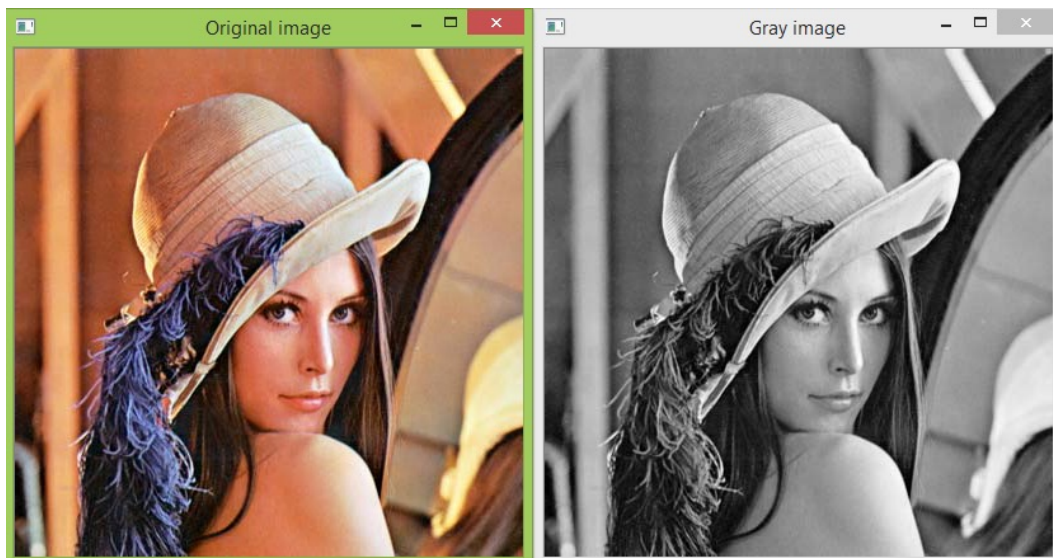
BEV Transform

02 BEV 주요 코드 설명

```
12 def bev_transform(image):  
13     # 이미지의 높이와 너비 추출  
14     height, width = image.shape[:2]  
15  
16     # ROI 설정 (원본 이미지에서 4점)  
17     src = np.float32([  
18         ①[700, 450],  
19         ②[880, 450],  
20         ③[20, height],  
21         ④[1240, height]  
22     ])  
23  
24     # 변환된 영역 설정 (출력 이미지에서 4점)  
25     out_size = 1000  
26     dst = np.float32([  
27         ①[0, 0],  
28         ②[out_size, 0],  
29         ③[0, out_size],  
30         ④[out_size, out_size]  
31     ])  
32  
33     # bev 변환 수행  
34     matrix = cv2.getPerspectiveTransform(src, dst)  
35     bev_img = cv2.warpPerspective(image, matrix, (out_size, out_size))  
36  
37     return bev_img
```



03 Filtering Filtering이란?



Filtering

- 이미지 픽셀값에 대해 특정 연산을 진행하여 톤에 변화를 주거나, 원하는 픽셀값만 남기는 기법
- 여러 기법들 중 GrayScale, HSV Masking 실습 예정
- opencv에서는 cv2.cvtColor(), cv2.inRange() 등의 함수들을 통해 구현 가능

03 Filtering Filtering 실습

```
wooseok@wooseok-17ZD90Q-GX5LK:~/ros2_ws$ ros2 run takeout_week5 filtering
```

filtering 코드 실행

\$ ros2 run takeout_week5 filtering



Original Image



Grayscale



White Mask



Yellow Mask



Combined Mask



Masked Image

03 Filtering 주요 코드 설명

```

11 # 이미지 필터링
12 def filter_image(image):
13     # grayscale 변환
14     gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
15
16     # hsv 변환
17     hsv = cv2.cvtColor(image, cv2.COLOR_BGR2HSV)
18
19     # 흰색 마스크
20     lower_white = np.array([0, 0, 155])
21     upper_white = np.array([137, 29, 255])
22     mask_white = cv2.inRange(hsv, lower_white, upper_white)
23
24     # 노란색 마스크
25     lower_yellow = np.array([0, 72, 162])
26     upper_yellow = np.array([30, 255, 255])
27     mask_yellow = cv2.inRange(hsv, lower_yellow, upper_yellow)
28
29     # 흰색 + 노란색 마스크
30     combined_mask = cv2.bitwise_or(mask_white, mask_yellow)
31
32     # 흰색 + 노란색 마스크 적용한 이미지
33     masked_image = cv2.bitwise_and(image, image, mask=combined_mask)
34
35     return gray, mask_white, mask_yellow, combined_mask, masked_image

```

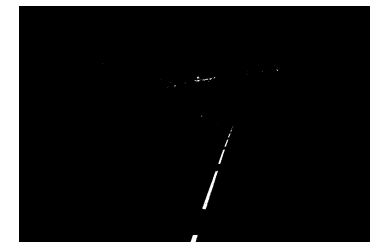
Grayscale



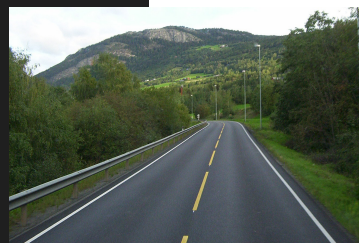
White Mask



Yellow Mask



Original Image



Combined Mask



Masked Image

03 Edge & Line Canny Edge란?

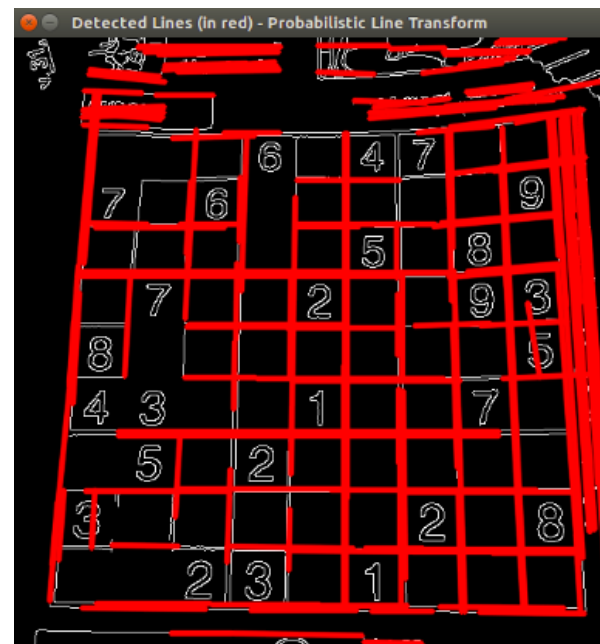


Canny Edge

- Edge는 이미지에서 픽셀값의 변화가 큰 부분(경계)을 의미
- Edge를 구하는 대표적인 알고리즘으로 Canny Edge가 있음
- Canny Edge는 multi-stage 알고리즘으로 총 4단계로 이루어져 있음
- opencv에서는 `cv2.Canny()` 함수를 통해 구현 가능

1. 노이즈 제거
2. 이미지에서 기울기 높은 부분 찾기
3. 최대값이 아닌 픽셀값을 0으로 만들기
4. 엣지 선별

04 Edge & Line Hough Line Transform이란?



Hough Line Transform

- Hough Line Transform은 이미지에서 직선을 찾아주는 변환 기법
- 주로 edge를 입력값으로 받음
- opencv에서는 cv2.HoughLines, cv2.HoughLinesP 등의 함수들을 통해 구현 가능
→ 이번 실습은 cv2.HoughLinesP 사용

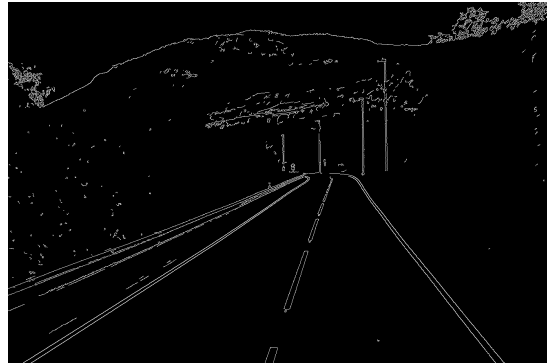
04 Edge & Line Edge & Line 실습

```
# edge_line 코드 실행  
$ ros2 run takeout_week5 edge_line
```

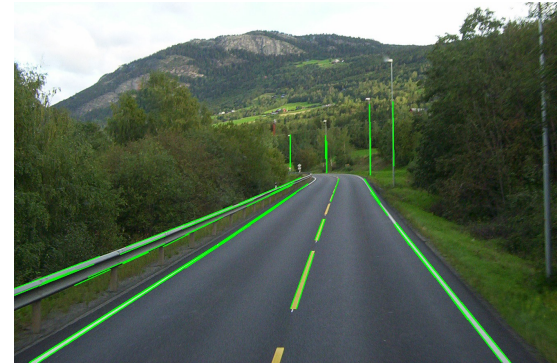
```
wooseok@wooseok-17ZD90Q-GX5LK:~/ros2_ws$ ros2 run takeout_week5 edge_line
```



Original Image



Canny Edges



Hough Lines

04 Edge & Line 주요 코드 설명

```

11 # canny edge & hoguh line 찾기
12 def detect_edges_and_lines(image):
13     # grayscale 변환
14     gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
15
16     # edge 추출
17     edges = cv2.Canny(gray, 150, 280)
18
19     # houghline 추출
20     lines = cv2.HoughLinesP(edges, 1, np.pi/180, 50, minLineLength=50, maxLineGap=3)
21
22     # hoguhline 그리기
23     output = image.copy()
24     if lines is not None:
25         for line in lines:
26             x1, y1, x2, y2 = line[0]
27             cv2.line(output, (x1, y1), (x2, y2), (0, 255, 0), 2)
28
29     return edges, output

```

tune these

```

void cv::Canny ( InputArray  image,
                 OutputArray edges,
                 double      threshold1,
                 double      threshold2,
                 int         apertureSize = 3,
                 bool        L2gradient = false )

```

```

void cv::HoughLinesP ( InputArray  image,
                      OutputArray lines,
                      double      rho,
                      double      theta,
                      int         threshold,
                      double      minLineLength = 0 ,
                      double      maxLineGap = 0
                      )

```

05 with Camera 카메라 실습

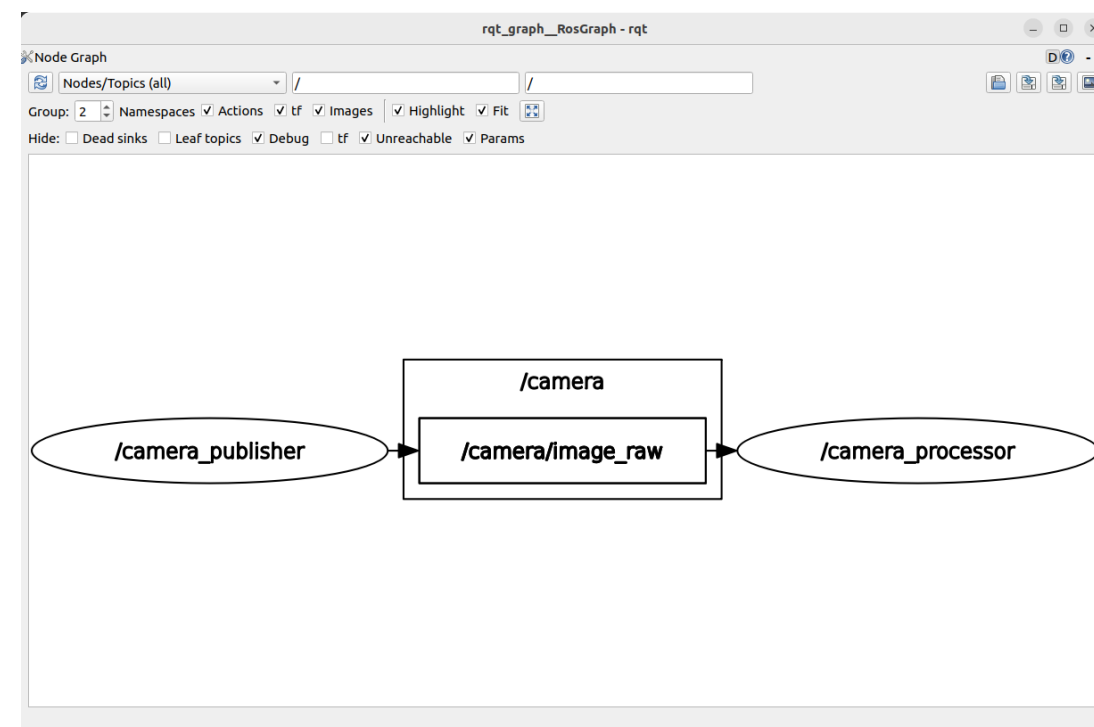
```
# 카메라 연결 후 아래 코드 실행
# 터미널 1
$ ros2 run takeout_week5 cam_pub
# 터미널 2 (새로운 터미널 창)
$ ros2 run takeout_week5 cam_sub

# 터미널 3 (새로운 터미널 창)
$ ros2 topic list
$ ros2 topic info /camera/image_raw
$ rqt_graph
```

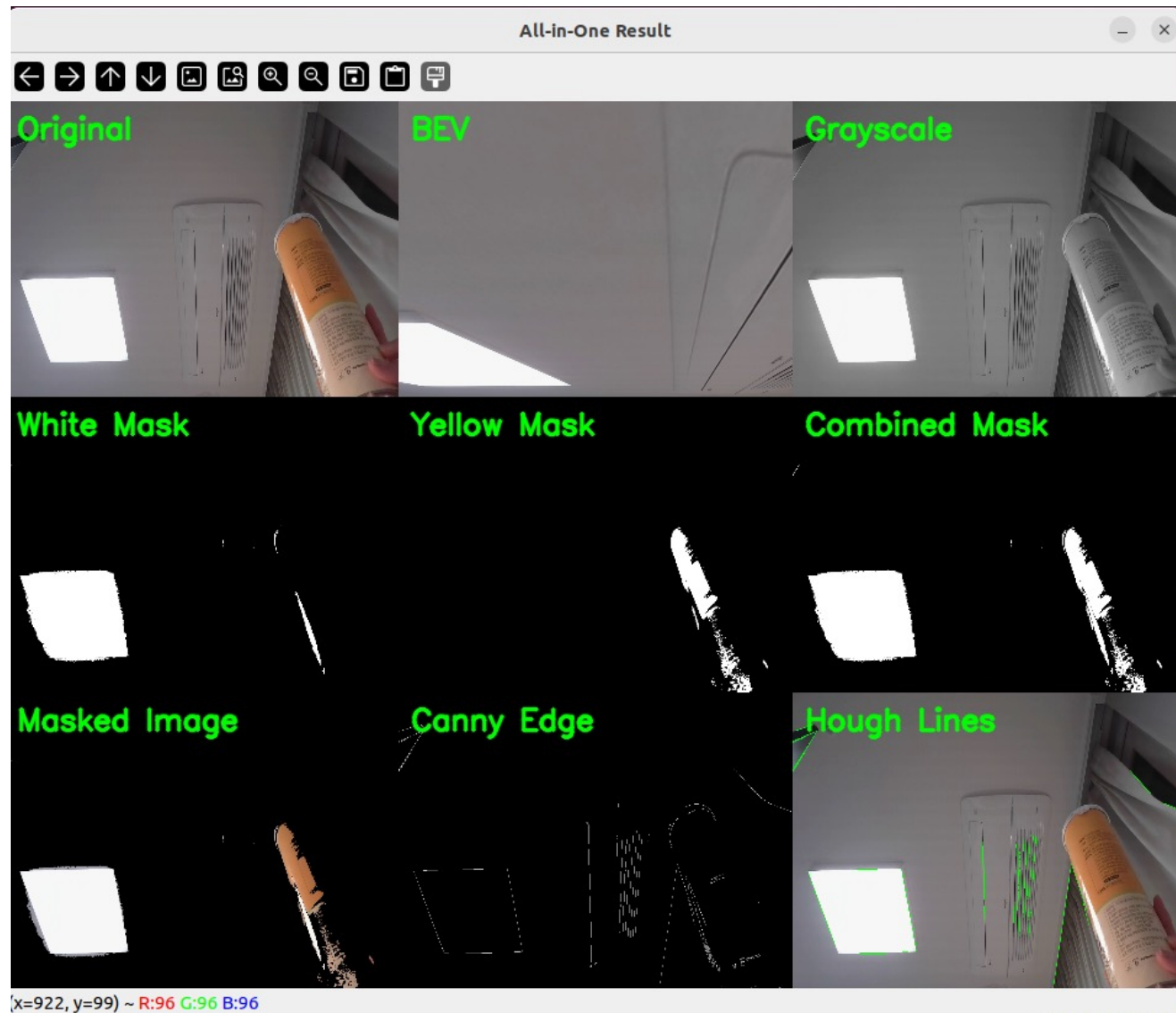
```
wooseok@wooseok-17ZD90Q-GX5LK: ~/ros2_ws 98x28
wooseok@wooseok-17ZD90Q-GX5LK:~/ros2_ws$ ros2 topic list
/camera/image_raw
/parameter_events
/rosout
wooseok@wooseok-17ZD90Q-GX5LK:~/ros2_ws$ ros2 topic info /camera/image_raw
Type: sensor_msgs/msg/Image
Publisher count: 1
Subscription count: 1
wooseok@wooseok-17ZD90Q-GX5LK:~/ros2_ws$ rqt_graph
```

```
wooseok@wooseok-17ZD90Q-GX5LK:~/ros2_ws$ ros2 run takeout_week5 cam_pub
[ WARN:0@0.529] global cap_gstreamer.cpp:1777 open OpenCV | GStreamer wa
rning: Cannot query video position: status=0, value=-1, duration=-1
```

```
wooseok@wooseok-17ZD90Q-GX5LK:~/ros2_ws$ ros2 run takeout_week5 cam_sub
```



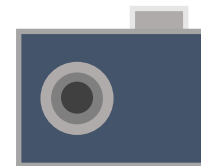
05 with Camera 카메라 실습



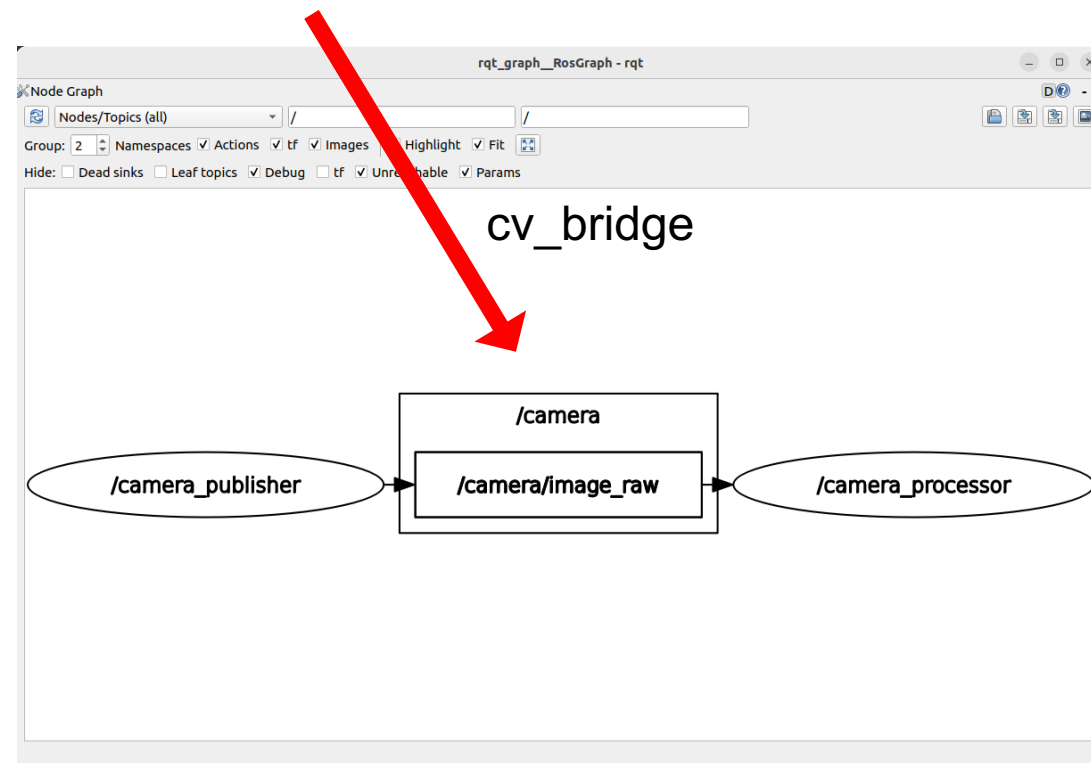
05 with Camera 주요 코드 설명

```
3 from sensor_msgs.msg import Image
4 from cv_bridge import CvBridge
```

```
7 # 카메라 프레임을 받아 카메라 토픽으로 publish 해주는 함수
8 class CameraPublisher(Node):
9     def __init__(self):
10         super().__init__('camera_publisher')
11         self.publisher_ = self.create_publisher(Image, '/camera/image_raw', 10)
12         self.bridge = CvBridge()
13         self.timer = self.create_timer(0.03, self.timer_callback) # 약 30 FPS
14         self.cap = cv2.VideoCapture(0)
15
16     def timer_callback(self):
17         ret, frame = self.cap.read()
18         if not ret:
19             self.get_logger().warning("카메라에서 프레임을 읽지 못했습니다.")
20             return
21         msg = self.bridge.cv2_to_imgmsg(frame, encoding='bgr8')
22         self.publisher_.publish(msg)
23
24     def destroy_node(self):
25         self.cap.release()
26         super().destroy_node()
```



Camera Frame

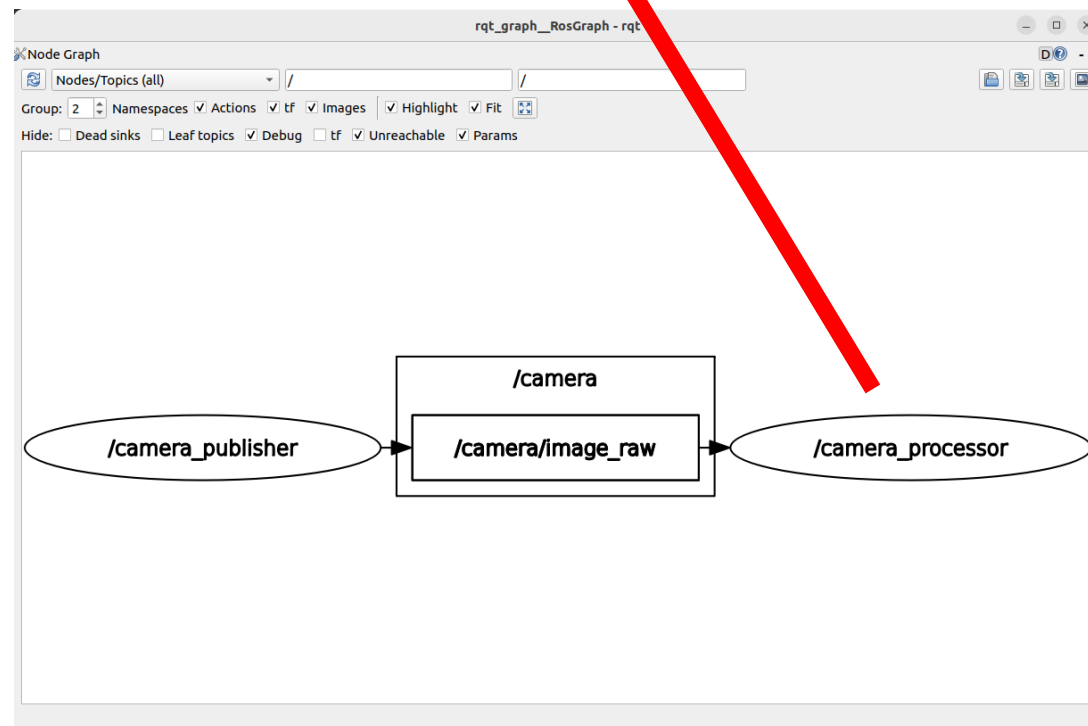


05 with Camera 주요 코드 설명

```
7 from takeout_week5.bev import bev_transform
8 from takeout_week5.filtering import filter_image
9 from takeout_week5.edge_line import detect_edges_and_lines
```

```
11 # 카메라 토픽을 받아 지금까지 한 모든 영상처리를 실시간으로 시각화하는 함수
12 class CameraProcessor(Node):
13     def __init__(self):
14         super().__init__('camera_processor')
15         self.subscriber = self.create_subscription(
16             Image,
17             '/camera/image_raw',
18             self.listener_callback,
19             10
20         )
21         self.bridge = CvBridge()
22
23     def listener_callback(self, msg):
24         frame = self.bridge.imgmsg_to_cv2(msg, desired_encoding='bgr8')
25         self.process_frame(frame)
26
27     # 영상처리
28     def process_frame(self, frame):
29         bev = bev_transform(frame)
30         gray, white_mask, yellow_mask, combined_mask, masked_image = filter_image(frame)
31         edges, hough = detect_edges_and_lines(frame)
32
33         row1 = self.stack_images_with_labels(
34             [frame, bev, gray],
35             ["Original", "BEV", "Grayscale"]
36         )
37         row2 = self.stack_images_with_labels(
38             [white_mask, yellow_mask, combined_mask],
39             ["White Mask", "Yellow Mask", "Combined Mask"]
40         )
41         row3 = self.stack_images_with_labels(
42             [masked_image, edges, hough],
43             ["Masked Image", "Canny Edge", "Hough Lines"]
44         )
45
46         final = np.vstack([row1, row2, row3])
47         cv2.imshow("All-in-One Result", final)
48         cv2.waitKey(1)
```

bev + filtering + edge_line



05 이미지 출처

- 슬라이드 3
 - <https://opencv.org/blog/learning-opencv/>
- 슬라이드 4
 - https://docs.opencv.org/4.x/d5/d69/tutorial_py_non_local_means.html
 - <https://developer-lionhong.tistory.com/28>
- 슬라이드 10
 - https://velog.io/@choonsik_mom/Bird-eye-view-%EA%B5%AC%ED%98%84
- 슬라이드 13
 - <https://techtutorialsx.com/2018/06/02/python-opencv-converting-an-image-to-gray-scale/>
 - <https://medium.com/@VinitKumarGupta/color-detection-in-opencv-a-hands-on-project-to-master-hsv-filtering-5fa9fd928561>
- 슬라이드 19
 - https://docs.opencv.org/4.x/dd/d1a/group_imgproc_feature.html#ga04723e007ed888ddf11d9ba04e2232de
 - https://docs.opencv.org/3.4/dd/d1a/group_imgproc_feature.html#ga8618180a5948286384e3b7ca02f6feeb
- 실습 이미지
 - https://ko.wiktionary.org/wiki/파일:Road_in_Norway.jpg



BRING OUT YOUR IDEA

TAKE OUT

문의처

19기	회장	정범교	010-4845-8209
24기	부회장	김성욱	010-5936-7623