



11차시 ROS2

vision_opencv/cv_bridge

[my_cv_bridge_demo.zip](#)

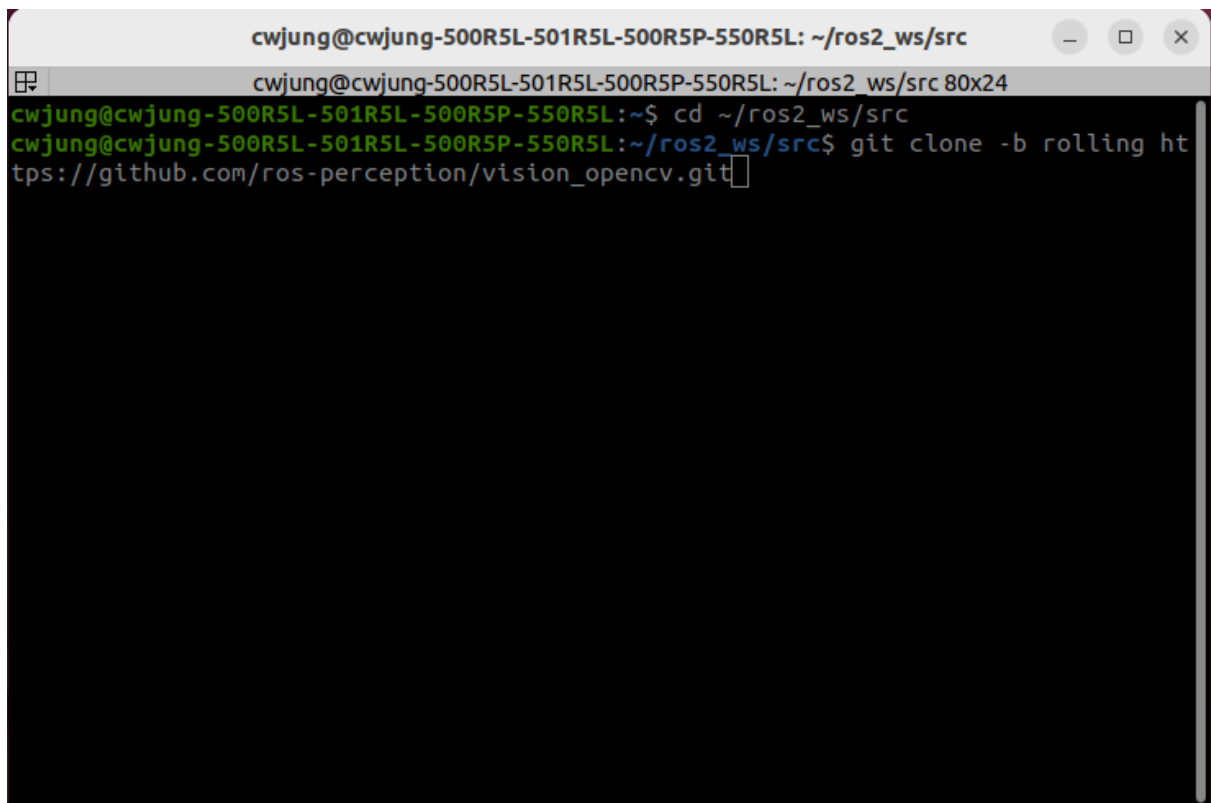
vision_opencv

- 개요
 - ROS2에서 OpenCV를 사용해 이미지 처리 및 CV 기능을 구현할 수 있도록 해주는 패키지 모음.
 - ROS 이미지 메시지와 OpenCV 이미지 간 변환
 - 카메라 모델을 통한 투영, 역투영 등 기하 정보 처리
 - OpenCV 기반 영상 처리 기능 test
- 구성요소
 - **cv_bridge 패키지: ROS 이미지 ↔ OpenCV 이미지 변환**
 - image_geometry: 카메라 모델 기반 3D 투영/역투영 도구
 - opencv_tests: vision_opencv 기능 검증용 테스트 노드들
- cv_bridge
 - ROS의 `sensor_msgs/msg/Image` 타입 ↔ OpenCV `cv::Mat` 이미지 간 변환
 - 주요 메서드
 - `imgmsg_to_cv2(msg, desired_encoding='bgr8')` : ROS 이미지 메시지를 OpenCV 이미지로 변환
 - `cv2_to_imgmsg(cv_image, encoding='bgr8')` : OpenCV 이미지를 ROS 이미지 메시지로 변환
- image_geometry

- ROS 카메라 파라미터 메시지를 활용한 카메라 모델 유틸리티
- 주요 클래스
 - `PinholeCameraModel` Intrinsic matrix, distortion params 등으로 투영/역투영 제공
 - `StereoCameraModel` 스테레오 카메라 처리 도구 제공
- 주요 기능
 - 이미지 좌표 ↔ 3D 공간 좌표 간 변환
 - ROI 보정, 카메라 왜곡 보정 등
- 응용
 - 깊이 이미지 활용 시
 - 3D 포인트를 픽셀 좌표로 투영하고 싶을 때
 - 카메라 캘리브레이션 정보를 활용한 정밀 위치 추정
- `opencv_tests`: 기능 테스트를 위한 샘플 노드

vision_opencv/cv_bridge 실습

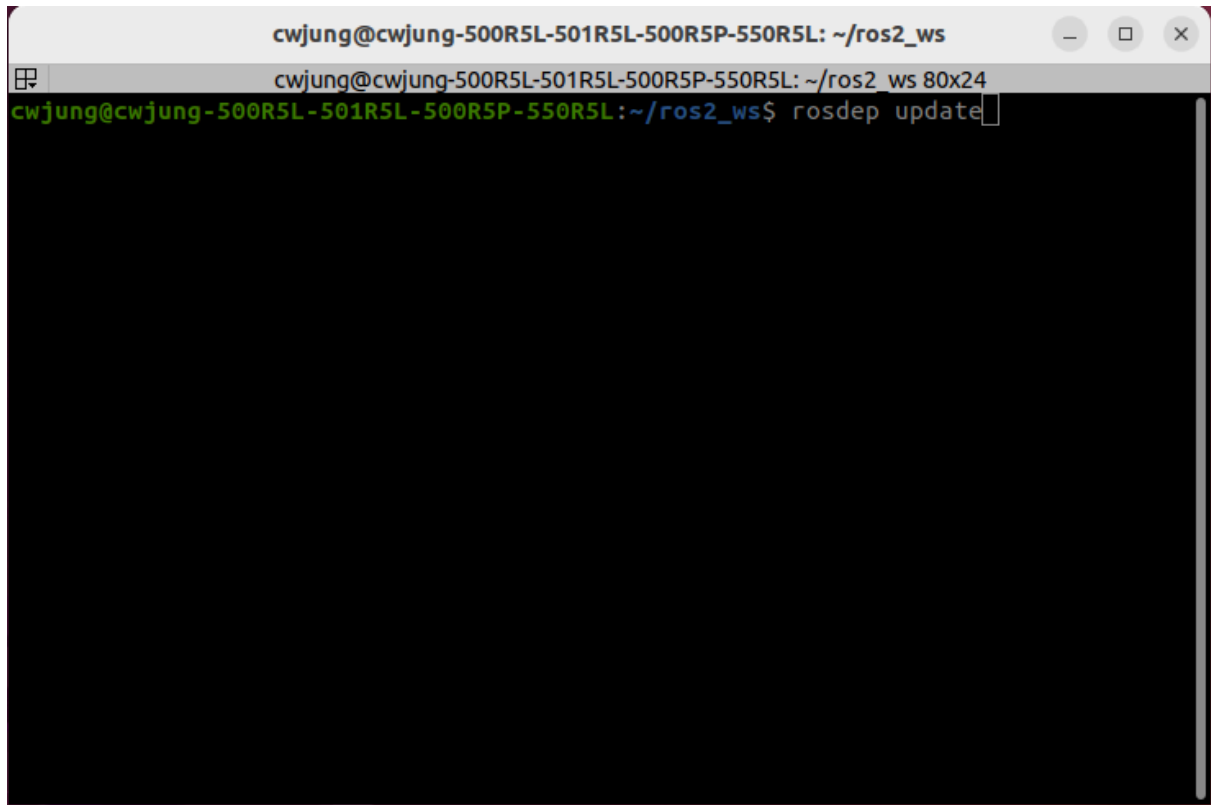
- 워크스페이스 이동 후 github에서 `ros-perception/vision_opencv` 패키지를 clone



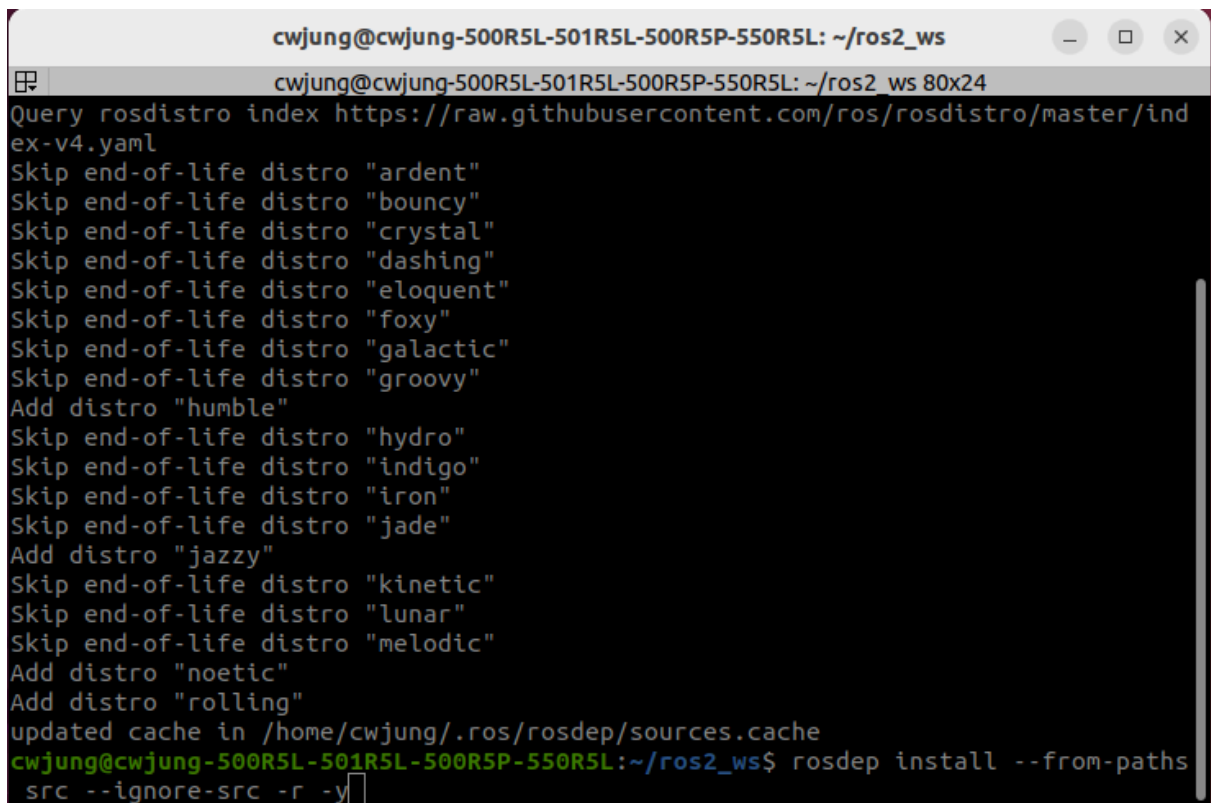
```

cwjung@cwjung-500R5L-501R5L-500R5P-550R5L: ~/ros2_ws/src
cwjung@cwjung-500R5L-501R5L-500R5P-550R5L: ~/ros2_ws/src 80x24
cwjung@cwjung-500R5L-501R5L-500R5P-550R5L:~$ cd ~/ros2_ws/src
cwjung@cwjung-500R5L-501R5L-500R5P-550R5L:~/ros2_ws/src$ git clone -b rolling ht
tps://github.com/ros-perception/vision_opencv.git
  
```

- rosdep을 활용한 의존성 설치



```
cwjung@cwjung-500R5L-501R5L-500R5P-550R5L: ~/ros2_ws
cwjung@cwjung-500R5L-501R5L-500R5P-550R5L: ~/ros2_ws 80x24
cwjung@cwjung-500R5L-501R5L-500R5P-550R5L:~/ros2_ws$ rosdep update
```



```
cwjung@cwjung-500R5L-501R5L-500R5P-550R5L: ~/ros2_ws
cwjung@cwjung-500R5L-501R5L-500R5P-550R5L: ~/ros2_ws 80x24
Query rosdistro index https://raw.githubusercontent.com/ros/rosdistro/master/index-v4.yaml
Skip end-of-life distro "ardent"
Skip end-of-life distro "bouncy"
Skip end-of-life distro "crystal"
Skip end-of-life distro "dashing"
Skip end-of-life distro "eloquent"
Skip end-of-life distro "foxy"
Skip end-of-life distro "galactic"
Skip end-of-life distro "groovy"
Add distro "humble"
Skip end-of-life distro "hydro"
Skip end-of-life distro "indigo"
Skip end-of-life distro "iron"
Skip end-of-life distro "jade"
Add distro "jazzy"
Skip end-of-life distro "kinetic"
Skip end-of-life distro "lunar"
Skip end-of-life distro "melodic"
Add distro "noetic"
Add distro "rolling"
updated cache in /home/cwjung/.ros/rosdep/sources.cache
cwjung@cwjung-500R5L-501R5L-500R5P-550R5L:~/ros2_ws$ rosdep install --from-paths
src --ignore-src -r -y
```

- vision_opencv 패키지 빌드 및 빌드 결과 적용(ROS2 Humble 환경)

```
cwjung@cwjung-500R5L-501R5L-500R5P-550R5L: ~/ros2_ws
cwjung@cwjung-500R5L-501R5L-500R5P-550R5L: ~/ros2_ws 80x24
cwjung@cwjung-500R5L-501R5L-500R5P-550R5L:~/ros2_ws$ colcon build --packages-select vision_opencv
Starting >>> vision_opencv
[0.543s] WARNING:colcon.colcon_core.shell:The following packages are in the work space but haven't been built:
- cv_bridge
- image_geometry
They are being used from the following locations instead:
- /opt/ros/humble
- /opt/ros/humble
To suppress this warning ignore these packages in the workspace:
--packages-ignore cv_bridge image_geometry
Finished <<< vision_opencv [0.92s]

Summary: 1 package finished [1.31s]
cwjung@cwjung-500R5L-501R5L-500R5P-550R5L:~/ros2_ws$ source install/setup.bash
cwjung@cwjung-500R5L-501R5L-500R5P-550R5L:~/ros2_ws$ ros2 pkg list | grep cv_bridge
cv_bridge
cwjung@cwjung-500R5L-501R5L-500R5P-550R5L:~/ros2_ws$
```

패키지 빌드를 통해 cv_bridge가 잘 설치된 것을 확인할 수 있다.

cv_bridge: ROS 이미지 ↔ OpenCV 이미지 변환

예시

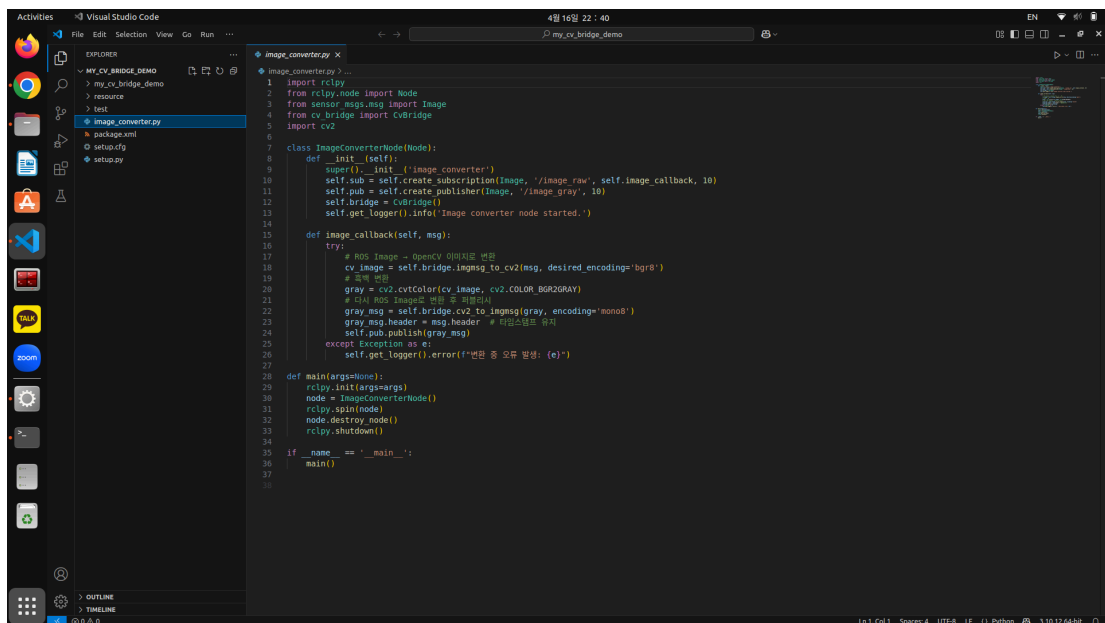
- 개요: ROS2 Humble + cv_bridge를 이용해서 OpenCV 이미지로 변환한 뒤 grayscale 처리한 이후 다시 publish하는 sample node 구현.
- 목표
 - sensor_msgs/msg/Image 타입의 ROS image topic을 subscribe
 - OpenCV 이미지를 그레이스케일로 변환
 - 이후 다시 ROS 이미지로 변환해서 publish
- my_cv_bridge_demo 패키지 생성

```
cwjung@cwjung-500R5L-501R5L-500R5P-550R5L: ~/ros2_ws/src
cwjung@cwjung-500R5L-501R5L-500R5P-550R5L: ~/ros2_ws/src 80x24
cwjung@cwjung-500R5L-501R5L-500R5P-550R5L:~/ros2_ws$ cd src
cwjung@cwjung-500R5L-501R5L-500R5P-550R5L:~/ros2_ws/src$ ros2 pkg create --build
-type ament_python my_cv_bridge_demo
going to create a new package
package name: my_cv_bridge_demo
destination directory: /home/cwjung/ros2_ws/src
package format: 3
version: 0.0.0
description: TODO: Package description
maintainer: ['cwjung <cwjung@todo.todo>']
licenses: ['TODO: License declaration']
build type: ament_python
dependencies: []
creating folder ./my_cv_bridge_demo
creating ./my_cv_bridge_demo/package.xml
creating source folder
creating folder ./my_cv_bridge_demo/my_cv_bridge_demo
creating ./my_cv_bridge_demo/setup.py
creating ./my_cv_bridge_demo/setup.cfg
creating folder ./my_cv_bridge_demo/resource
creating ./my_cv_bridge_demo/resource/my_cv_bridge_demo
creating ./my_cv_bridge_demo/my_cv_bridge_demo/__init__.py
creating folder ./my_cv_bridge_demo/test
creating ./my_cv_bridge_demo/test/test_copyright.py
```

- my_cv_bridge_demo 패키지 구성요소

- image_converter.py

cv_bridge를 사용해서 /image를 구독하고, gray 이미지로 변환한 후 /image_gray로 퍼블리시하는 노드.



```
1 import rclpy
2 from rclpy.node import Node
3 from sensor_msgs.msg import Image
4 from cv_bridge import CvBridge
5 import cv2
6
7 class ImageConverterNode(Node):
8     def __init__(self):
9         super().__init__('image_converter')
10        self.sub = self.create_subscription(Image, '/image_raw', self.image_callback, 10)
11        self.pub = self.create_publisher(Image, '/image_gray', 10)
12        self.bridge = CvBridge()
13        self.get_logger().info('Image converter node started.')
14
15    def image_callback(self, msg):
16        try:
17            # ROS Image -> OpenCV 이미지로 변환
18            cv_image = self.bridge.imgmsg_to_cv2(msg, desired_encoding='bgr8')
19            # 컬러 변환
20            gray = cv2.cvtColor(cv_image, cv2.COLOR_BGR2GRAY)
21            # 다시 ROS Image로 변환 후 퍼블리시
22            gray_msg = self.bridge.cv2_to_imgmsg(gray, encoding='mono8')
23            gray_msg.header = msg.header # 타임스탬프 유지
24            self.pub.publish(gray_msg)
25        except Exception as e:
26            self.get_logger().error(f'변환 중 오류 발생: {e}')
27
28 def main(args=None):
29     rclpy.init(args=args)
30     node = ImageConverterNode()
31     rclpy.spin(node)
32     node.destroy_node()
33     rclpy.shutdown()
34
35 if __name__ == '__main__':
36     main()
37
```

```

import rclpy
from rclpy.node import Node
from sensor_msgs.msg import Image
from cv_bridge import CvBridge
import cv2

class ImageConverterNode(Node):
    def __init__(self):
        super().__init__('image_converter')
        #입력 토픽: /image (sensor_msgs/msg/Image)으로부터 원본 RGB 이미지
        self.sub = self.create_subscription(Image, '/image', self.image_callback, 10)
        #출력 토픽: /image_gray(grayscale 이미지) 전송
        #queue size=10: topic buffer queue 설정
        self.pub = self.create_publisher(Image, '/image_gray', 10)
        #ROS↔OpenCV 이미지 변환기
        self.bridge = CvBridge()
        self.get_logger().info('Image converter node started.')

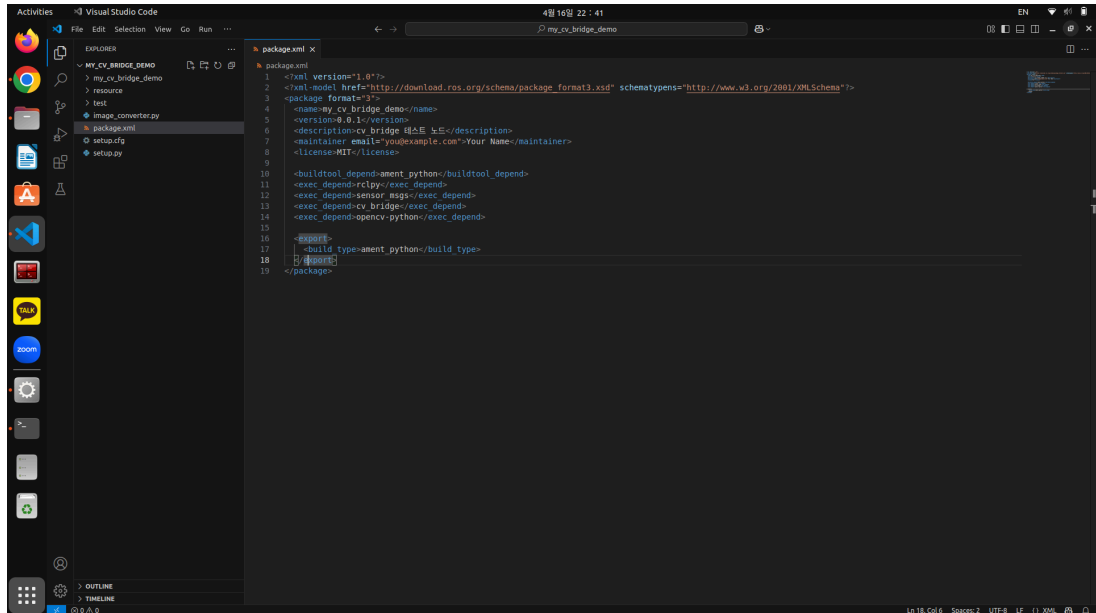
    def image_callback(self, msg):
        try:
            # ROS Image → OpenCV 이미지로 변환
            cv_image = self.bridge.imgmsg_to_cv2(msg, desired_encoding='bgr')
            # 흑백 변환
            gray = cv2.cvtColor(cv_image, cv2.COLOR_BGR2GRAY)
            # OpenCV 이미지를 다시 ROS Image로 변환 후 퍼블리시
            gray_msg = self.bridge.cv2_to_imgmsg(gray, encoding='mono8')
            gray_msg.header = msg.header # 타임스탬프 유지
            self.pub.publish(gray_msg) #/image_gray 토픽으로 발행
        except Exception as e:
            self.get_logger().error(f"변환 중 오류 발생: {e}")

def main(args=None):
    rclpy.init(args=args)
    node = ImageConverterNode()
    rclpy.spin(node)
    node.destroy_node()
    rclpy.shutdown()

```

```
if __name__ == '__main__':
    main()
```

○ package.xml



기존 package.xml에 아래와 같은 의존성을 추가하였다.

```
<exec_depend>rcpp</exec_depend>
<exec_depend>sensor_msgs</exec_depend>
<exec_depend>cv_bridge</exec_depend>
<exec_depend>opencv-python</exec_depend>
```

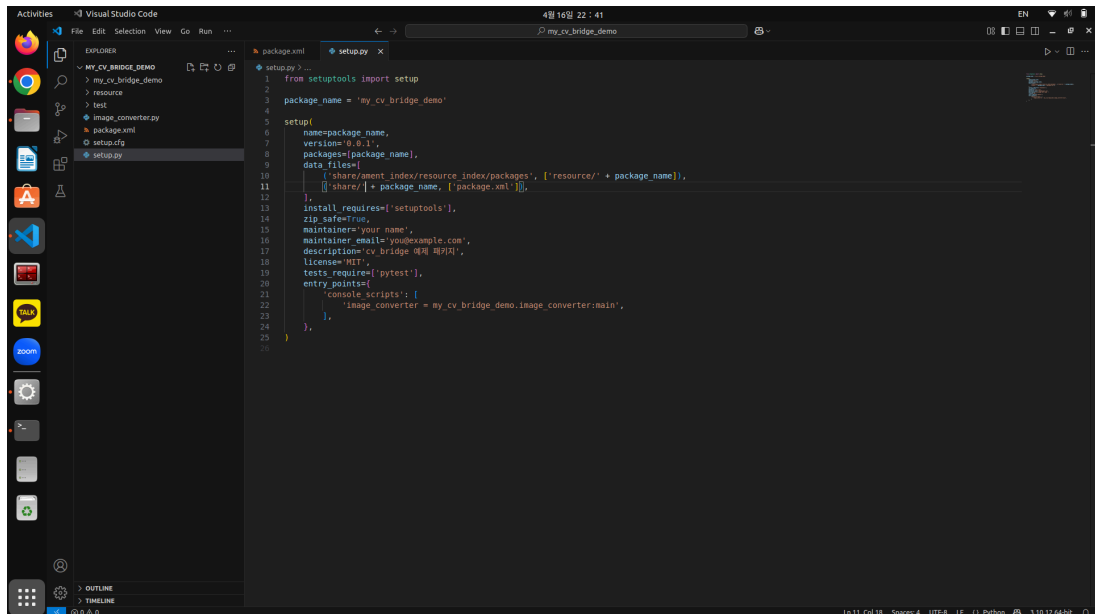
```
<?xml-model href="http://download.ros.org/schema/package_format3.xsd" schematypens="http://www.w3.org/2001/XMLSchema"?>
<package format="3">
  <name>my_cv_bridge_demo</name>
  <version>0.0.1</version>
  <description>cv_bridge 테스트 노드</description>
  <maintainer email="you@example.com">Your Name</maintainer>
  <license>MIT</license>
  <buildtool_depend>ament_python</buildtool_depend>
  <exec_depend>rcpp</exec_depend>
  <exec_depend>sensor_msgs</exec_depend>
  <exec_depend>cv_bridge</exec_depend>
```

```

<exec_depend>opencv-python</exec_depend>
<export>
  <build_type>ament_python</build_type>
</export>
</package>

```

◦ setup.py



```
from setuptools import setup
```

```
package_name = 'my_cv_bridge_demo'
```

```

setup(
    name=package_name,
    version='0.0.1',
    packages=[package_name],
    data_files=[
        ('share/ament_index/resource_index/packages', ['resource/' + pa
        ('share/' + package_name, ['package.xml']),
    ],
    install_requires=['setuptools'],
    zip_safe=True,
    maintainer='your name',
    maintainer_email='you@example.com',

```



```

description='cv_bridge 예제 패키지',
license='MIT',
tests_require=['pytest'],
entry_points={
    'console_scripts': [
        'image_converter = my_cv_bridge_demo.image_converter:main
    ],
},
)

```

setup.py의 entry_points에서 image_converter: 실행 이름,
my_cv_bridge_demo.image_converter:main은 main() 함수가 들어있는 위치.

```

entry_points={
    'console_scripts': [
        'image_converter = my_cv_bridge_demo.image_converter:main',
    ],
},

```

- my_cv_bridge_demo 패키지 빌드(오류 발생)

```

cwjung@cwjung-500R5L-501R5L-500R5P-550R5L: ~/ros2_ws
cwjung@cwjung-500R5L-501R5L-500R5P-550R5L: ~/ros2_ws 80x24
creating ./my_cv_bridge_demo/resource/my_cv_bridge_demo
creating ./my_cv_bridge_demo/my_cv_bridge_demo/__init__.py
creating folder ./my_cv_bridge_demo/test
creating ./my_cv_bridge_demo/test/test_copyright.py
creating ./my_cv_bridge_demo/test/test_flake8.py
creating ./my_cv_bridge_demo/test/test_pep257.py

[WARNING]: Unknown license 'TODO: License declaration'. This has been set in the
package.xml, but no LICENSE file has been created.
It is recommended to use one of the ament license identifiers:
Apache-2.0
BSL-1.0
BSD-2.0
BSD-2-Clause
BSD-3-Clause
GPL-3.0-only
LGPL-3.0-only
MIT
MIT-0
cwjung@cwjung-500R5L-501R5L-500R5P-550R5L:~/ros2_ws/src$ touch my_cv_bridge_demo
/image_converter.py
cwjung@cwjung-500R5L-501R5L-500R5P-550R5L:~/ros2_ws/src$ cd ..
cwjung@cwjung-500R5L-501R5L-500R5P-550R5L:~/ros2_ws$ colcon build --packages-select
my_cv_bridge_demo

```

```

cwjung@cwjung-500R5L-501R5L-500R5P-550R5L: ~/ros2_ws
Finished <<< my_cv_bridge_demo [1.25s]

Summary: 1 package finished [1.84s]
cwjung@cwjung-500R5L-501R5L-500R5P-550R5L:~/ros2_ws$ source install/setup.bash
cwjung@cwjung-500R5L-501R5L-500R5P-550R5L:~/ros2_ws$ ros2 run my_cv_bridge_demo
image_converter
Traceback (most recent call last):
  File "/home/cwjung/ros2_ws/install/my_cv_bridge_demo/lib/my_cv_bridge_demo/ima
ge_converter", line 33, in <module>
    sys.exit(load_entry_point('my-cv-bridge-demo==0.0.1', 'console_scripts', 'im
age_converter'))()
  File "/home/cwjung/ros2_ws/install/my_cv_bridge_demo/lib/my_cv_bridge_demo/ima
ge_converter", line 25, in importlib_load_entry_point
    return next(matches).load()
  File "/usr/lib/python3.10/importlib/metadata/__init__.py", line 171, in load
module = import_module(match.group('module'))
  File "/usr/lib/python3.10/importlib/__init__.py", line 126, in import_module
    return _bootstrap._gcd_import(name[level:], package, level)
  File "<frozen importlib._bootstrap>", line 1050, in _gcd_import
  File "<frozen importlib._bootstrap>", line 1027, in _find_and_load
  File "<frozen importlib._bootstrap>", line 1004, in _find_and_load_unlocked
ModuleNotFoundError: No module named 'my_cv_bridge_demo.image_converter'
[ros2run]: Process exited with failure 1
cwjung@cwjung-500R5L-501R5L-500R5P-550R5L:~/ros2_ws$
```

오류를 해결하고자 my_cv_bridge_demo 패키지 내 구조를 확인해보았다. 확인해보니 패키지 구조 내 image_converter.py 파일의 위치가 올바르지 않아 올바른 디렉토리로 다시 옮겨주었다.

```

ros2_ws/src/my_cv_bridge_demo/
├── my_cv_bridge_demo/
│   ├── init.py
│   └── image_converter.py
├── package.xml
└── setup.py
```

<오류 발생한 이후 디렉토리 위치 수정한 패키지 구조>

```

cwjung@cwjung-500R5L-501R5L-500R5P-550R5L: ~/ros2_ws/src/my_cv_bridge_demo
├── setup.py
├── test
│   ├── test_copyright.py
│   ├── test_flake8.py
│   └── test_pep257.py
└── 3 directories, 9 files
cwjung@cwjung-500R5L-501R5L-500R5P-550R5L:~/ros2_ws/src/my_cv_bridge_demo$ tree
.
├── my_cv_bridge_demo
│   ├── image_converter.py
│   └── __init__.py
├── package.xml
├── resource
│   └── my_cv_bridge_demo
├── setup.cfg
├── setup.py
├── test
│   ├── test_copyright.py
│   ├── test_flake8.py
│   └── test_pep257.py
└── 3 directories, 9 files
cwjung@cwjung-500R5L-501R5L-500R5P-550R5L:~/ros2_ws/src/my_cv_bridge_demo$
```

이후 패키지 빌드를 다시 실행하니 성공적으로 빌드가 되었다.

```

cwjung@cwjung-500R5L-501R5L-500R5P-550R5L: ~/ros2_ws
├── cwjung@cwjung-500R5L-501R5L-500R5P-550R5L: ~/ros2_ws 80x24
├── cwjung@cwjung-500R5L-501R5L-500R5P-550R5L:~/ros2_ws/src/my_cv_bridge_demo$ cd ~/ros2_ws
├── cwjung@cwjung-500R5L-501R5L-500R5P-550R5L:~/ros2_ws$ colcon build --packages-select my_cv_bridge_demo
├── Starting >>> my_cv_bridge_demo
├── [0.547s] WARNING:colcon.colcon_core.shell:The following packages are in the workspace but haven't been built:
├── - cv_bridge
├── They are being used from the following locations instead:
├── - /opt/ros/humble
├── To suppress this warning ignore these packages in the workspace:
├── --packages-ignore cv_bridge
├── Finished <<< my_cv_bridge_demo [1.21s]
├── Summary: 1 package finished [1.59s]
├── cwjung@cwjung-500R5L-501R5L-500R5P-550R5L:~/ros2_ws$ source install/setup.bash
├── cwjung@cwjung-500R5L-501R5L-500R5P-550R5L:~/ros2_ws$ ros2 run my_cv_bridge_demo image_converter
└──
```

- 테스트

1. `$ros2 run image_tools cam2image`

image_tools 패키지의 cam2image로 publish하여 /image라는 topic으로 이미지를 publish 한다.

2. `$ros2 run my_cv_bridge_demo image_converter`

image_converter node 실행

3. `$ros2 topic echo /image_gray`

흑백으로 변환된 이미지를 확인한다.

4. `$rqt`

rqt를 실행하여 웹캠 상으로 비춰지는 이미지의 흑백 처리가 잘 이루어지는지 확인한다.

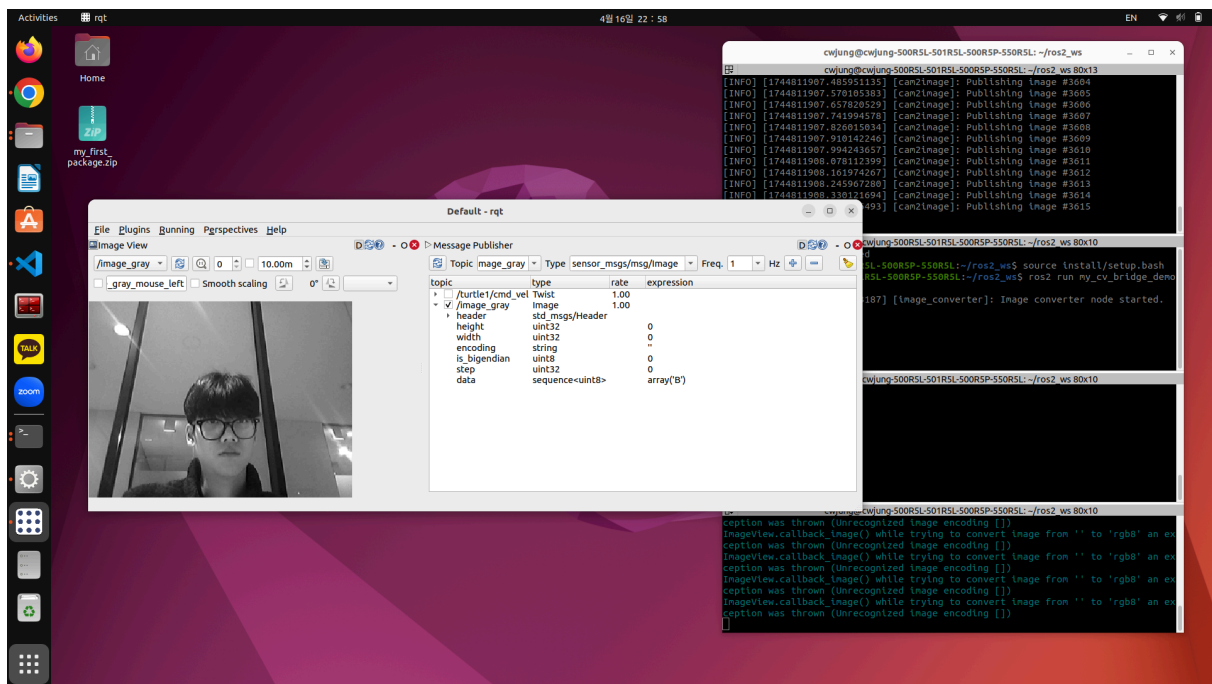
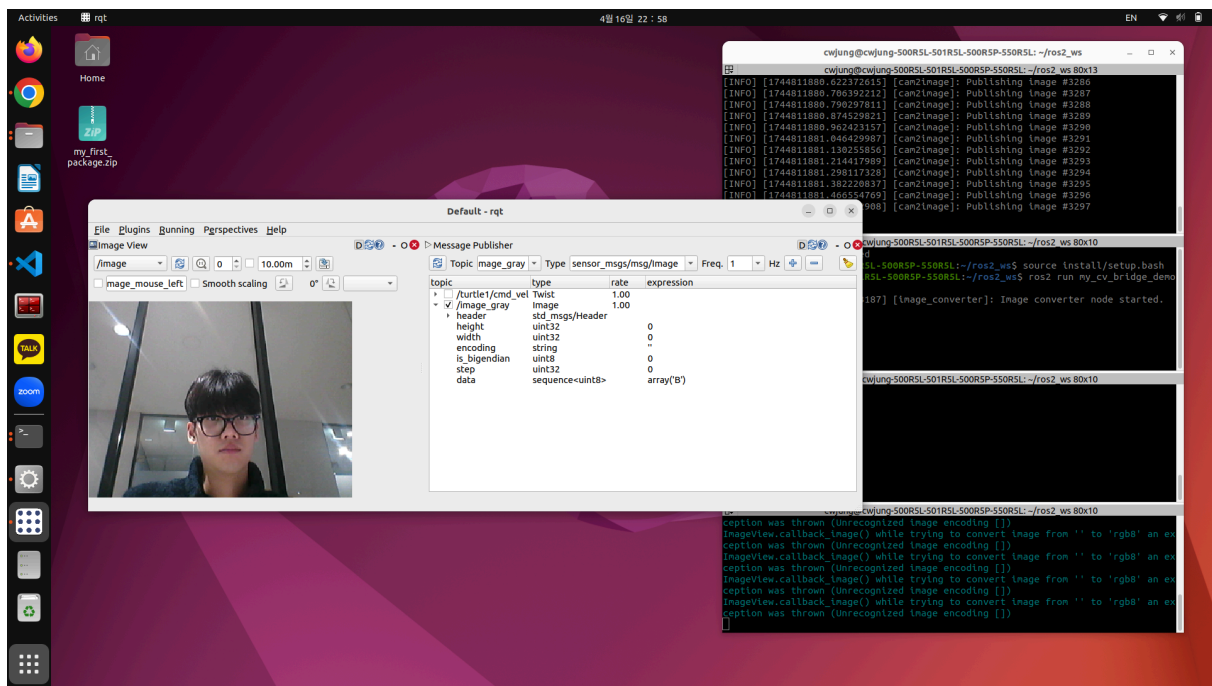
```

cwjung@cwjung-500R5L-501R5L-500R5P-550R5L: ~/ros2_ws
cwjung@cwjung-500R5L-501R5L-500R5P-550R5L: ~/ros2_ws 80x13
[INFO] [1744811916.694041951] [cam2image]: Publishing image #3713
[INFO] [1744811916.777916807] [cam2image]: Publishing image #3714
[INFO] [1744811916.865954164] [cam2image]: Publishing image #3715
[INFO] [1744811916.950147799] [cam2image]: Publishing image #3716
[INFO] [1744811917.034044041] [cam2image]: Publishing image #3717
[INFO] [1744811917.118006427] [cam2image]: Publishing image #3718
[INFO] [1744811917.201962703] [cam2image]: Publishing image #3719
[INFO] [1744811917.285915999] [cam2image]: Publishing image #3720
[INFO] [1744811917.370030980] [cam2image]: Publishing image #3721
^C[INFO] [1744811917.405329768] [rclcpp]: signal_handler(signum=2)
[INFO] [1744811917.453958931] [cam2image]: Publishing image #3722
cwjung@cwjung-500R5L-501R5L-500R5P-550R5L:~/ros2_ws$ ros2 run image_tools cam2im
age
cwjung@cwjung-500R5L-501R5L-500R5P-550R5L:~/ros2_ws 80x10
File "/opt/ros/humble/local/lib/python3.10/dist-packages/rclpy/executors.py",
line 723, in wait_for_ready_callbacks
    return next(self._cb_iter)
File "/opt/ros/humble/local/lib/python3.10/dist-packages/rclpy/executors.py",
line 620, in _wait_for_ready_callbacks
    wait_set.wait(timeout_nsec)
KeyboardInterrupt
[ros2run]: Interrupt
cwjung@cwjung-500R5L-501R5L-500R5P-550R5L:~/ros2_ws$ ros2 run my_cv_bridge_demo
image_converter
cwjung@cwjung-500R5L-501R5L-500R5P-550R5L:~/ros2_ws 80x10
frame_id: ''
height: 0
width: 0
encoding: ''
is_bigendian: 0
step: 0
data: []
---
^Ccwjung@cwjung-500R5L-501R5L-500R5P-550R5L:~/ros2_ws$ ros2 topic echo /image_gr
ay
cwjung@cwjung-500R5L-501R5L-500R5P-550R5L:~/ros2_ws 80x10
ImageView.callback_image() while trying to convert image from '' to 'rgb8' an ex
ception was thrown (Unrecognized image encoding [])
ImageView.callback_image() while trying to convert image from '' to 'rgb8' an ex
ception was thrown (Unrecognized image encoding [])
ImageView.callback_image() while trying to convert image from '' to 'rgb8' an ex
ception was thrown (Unrecognized image encoding [])
ImageView.callback_image() while trying to convert image from '' to 'rgb8' an ex
ception was thrown (Unrecognized image encoding [])
^C[INFO] [1744811932.551875478] [rclcpp]: signal_handler(signum=2)
cwjung@cwjung-500R5L-501R5L-500R5P-550R5L:~/ros2_ws$ rqt

```

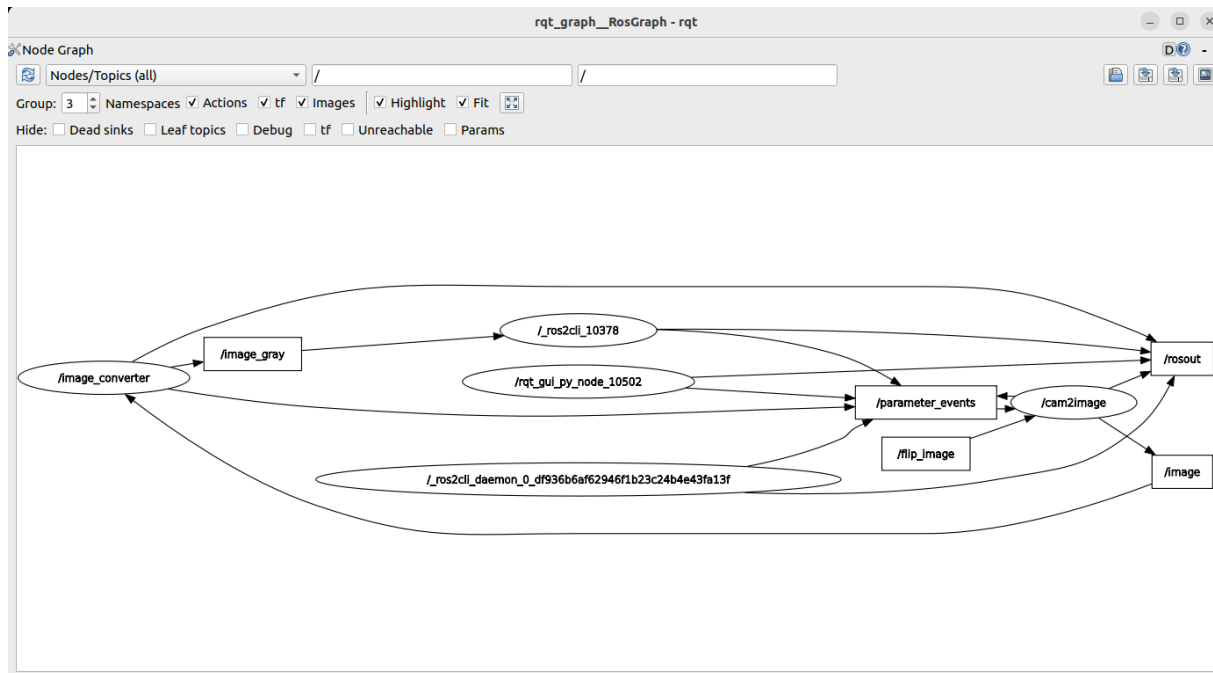
- rqt 실행 결과

rqt에서 plugins→visualization→image view 선택하여 기존 이미지 토픽과 흑백 처리된 토픽을 선택하여 실시간으로 표시되는 것을 확인한다.



- rqt_graph

node간 topic 연결 상태 확인



- 노드/토픽 구조 정리
 - `/image_converter` 노드
 - `/image` topic을 구독하고 opencv를 이용해서 grayscale로 변환 후 `/image_gray`로 publish
 - 입력 토픽: `/image` (카메라)
 - 출력 토픽: `/image_gray` (변환된 이미지)
 - `/image_gray` 토픽
 - `image_converter`에서 publish
 - 다른 노드 또는 rqt GUI에서 구독이 가능함
 - `rqt_gui_py_node_10502`
 - `rqt_image_view` GUI에서 사용하는 내부 노드
 - `/parameter_events`와 `/rosout`을 publish 하고 있다.
 - `/parameter_events` 토픽: 노드의 파라미터들이 변경될 때마다 자동으로 이벤트를 퍼블리시하는 시스템 토픽.
 - 어떤 노드의 파라미터가 선언, 삭제, 변경될때 이벤트 발생
 - 파라미터를 가진 모든 노드가 기본적으로 publisher가 됨
 - `cam2image` 노드 = test용 이미지 송출 노드

- 카메라 이미지 데이터를 /image topic으로 publish함
- 출력 토픽: /image (원본 이미지)
- /rosout 토픽: 모든 노드의 로그 메시지를 publish 하는 중앙화된 시스템 토픽.
 - RCLCPP_INFO(), RCLCPP_ERROR() 같은 로그 함수 호출되면 로그가 /rosout 토픽으로 publish 됨.
 - 시스템 전체 로그 모니터링 가능.
- 요약

/cam2image —▶ /image —▶ /image_converter —▶ /image_gray —▶ /rqt_gui_py_node_10502