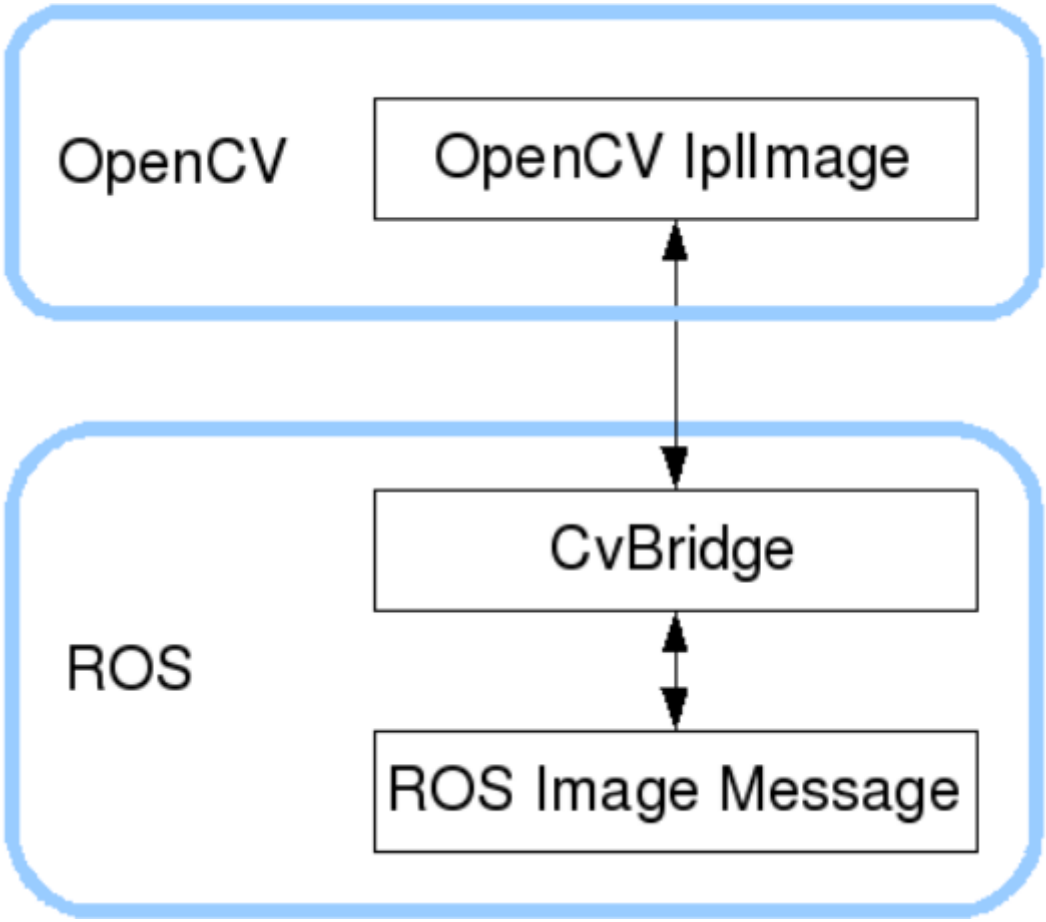
2. ROS+opencv application

This lesson takes USB driver-free camera as an example

ROS transmits images in its own sensor_msgs/Image message format and cannot directly perform image processing, but the provided [CvBridge] can perfectly convert and be converted image data formats. [CvBridge] is a ROS library, equivalent to the bridge between ROS and Opencv.

Opencv and ROS image data conversion is shown in the figure below:



2.1. Subscribe to image data and then publish the converted image data.

Run command

```
#Run the publish image topic data node
ros2 run yahboomcar_visual pub_image

#Choose one of the following nodes
#Run usb camera topic node (with image display)
ros2 launch usb_cam demo_launch.py

#Run the usb camera topic node (without image display)
ros2 launch usb_cam Demo_launch.py
```

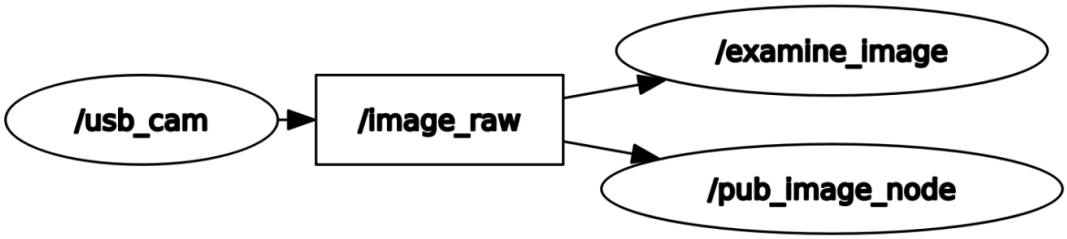
Relevant Notes:

1. Before entering the docker container, first confirm whether the USB driver-free camera device is connected, and add the camera device in the startup script. For specific tutorials, please see [Docker usage tutorial - 5. Enter the docker container]
2. Since the performance of some motherboards is not very good, the GUI display will be very stuck when running the USB camera topic node (such as jetson nano). You can run the node without image display, and then use the rqt_image_view tool in the tutorial below to view the image.

View node communication diagram

docker terminal input

```
ros2 run rqt_graph rqt_graph
```

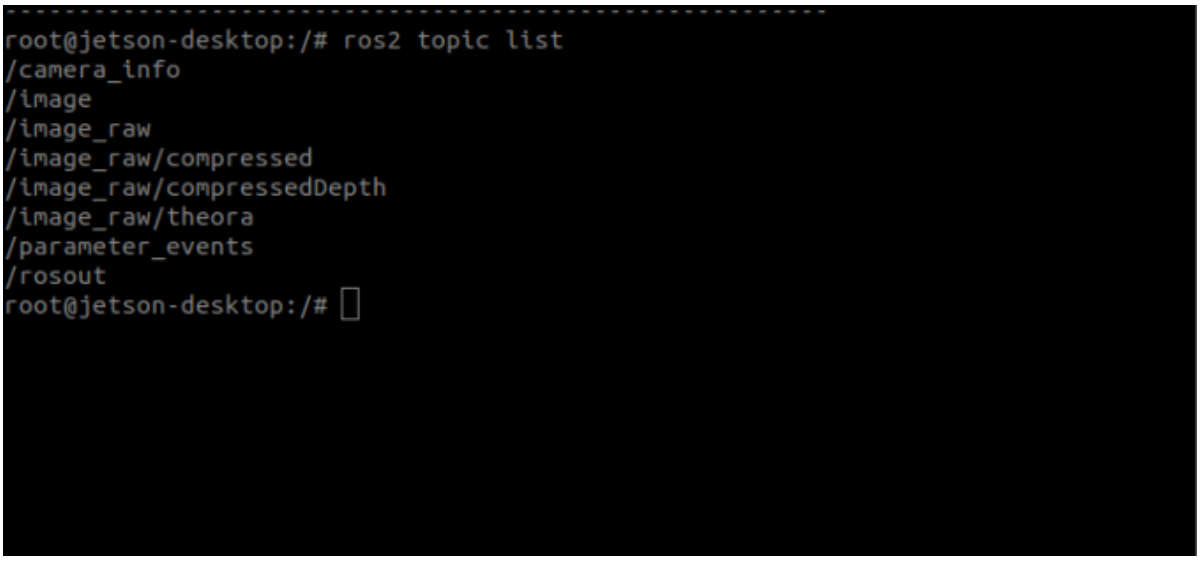


When running the node command without image display, the /examine_Image node is not displayed in rqt_graph

View topic data

First check which image topics are published, and enter it in the docker terminal.

```
ros2 topic list
```

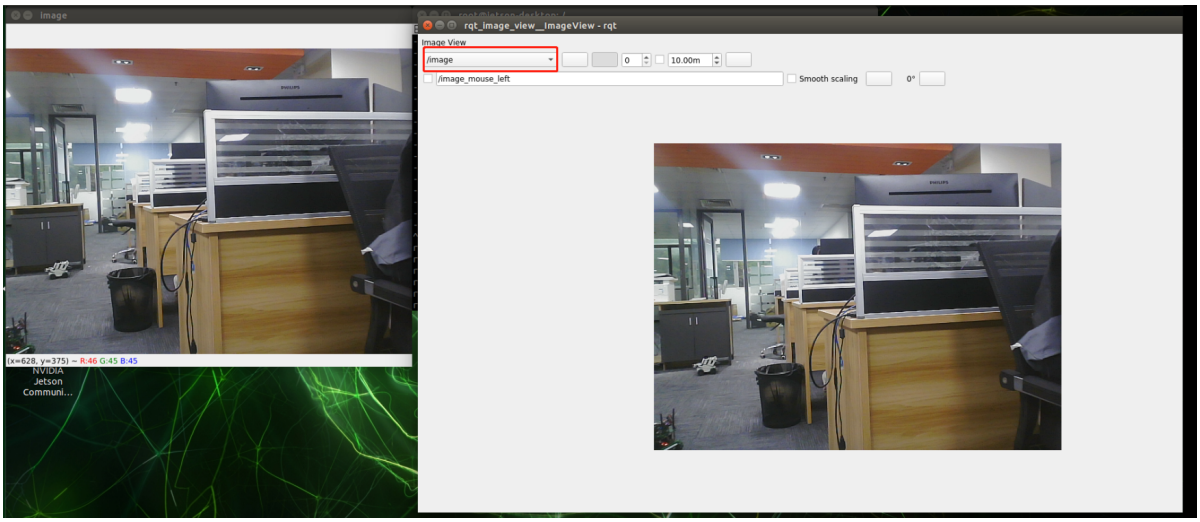


The /image is the topic data we published. Use the following command to print and see the data content of this topic.

```
ros2 topic echo /image
```

You can use the rqt_image_view tool to view images,

```
ros2 run rqt_image_view rqt_image_view
```



After opening, select the topic name/image in the upper left corner to view the image (sometimes it cannot be loaded for the first time after selecting the topic. You can end the process and re-enter the command to enter the interface to display the image)

2.2. Core code analysis

code path,

```
/root/yahboomcar_ros2_ws/yahboomcar_ws/src/yahboomcar_visual/yahboomcar_visual/pub_image.py
```

The implementation steps are roughly the same as the previous two. The program first subscribes to the topic data of /image_raw, and then converts it into image data. However, it also performs lattice conversion here to convert the image data into topic data, and then publishes it, which is the image topic data. ->Image data->Image topic data.

```
#导入opecv库以及cv_bridge库
import cv2 as cv
from cv_bridge import CvBridge
#创建CvBridge对象
self.bridge = CvBridge()
#定义一个订阅者订阅usb图像话题数据
self.sub_img = self.create_subscription(Image,'/image_raw',self.handleTopic,500)
#定义了图像话题数据发布者
self.pub_img = self.create_publisher(Image,'/image',500)
#msg转换成图像数据imgmsg_to_cv2，这里的bgr8是图像编码格式
frame = self.bridge.imgmsg_to_cv2(msg, "bgr8")
#图像数据转换的图像话题数据(cv2_to_imgmsg)然后发布出去
msg = self.bridge.cv2_to_imgmsg(frame, "bgr8")
self.pub_img.publish(msg)
```