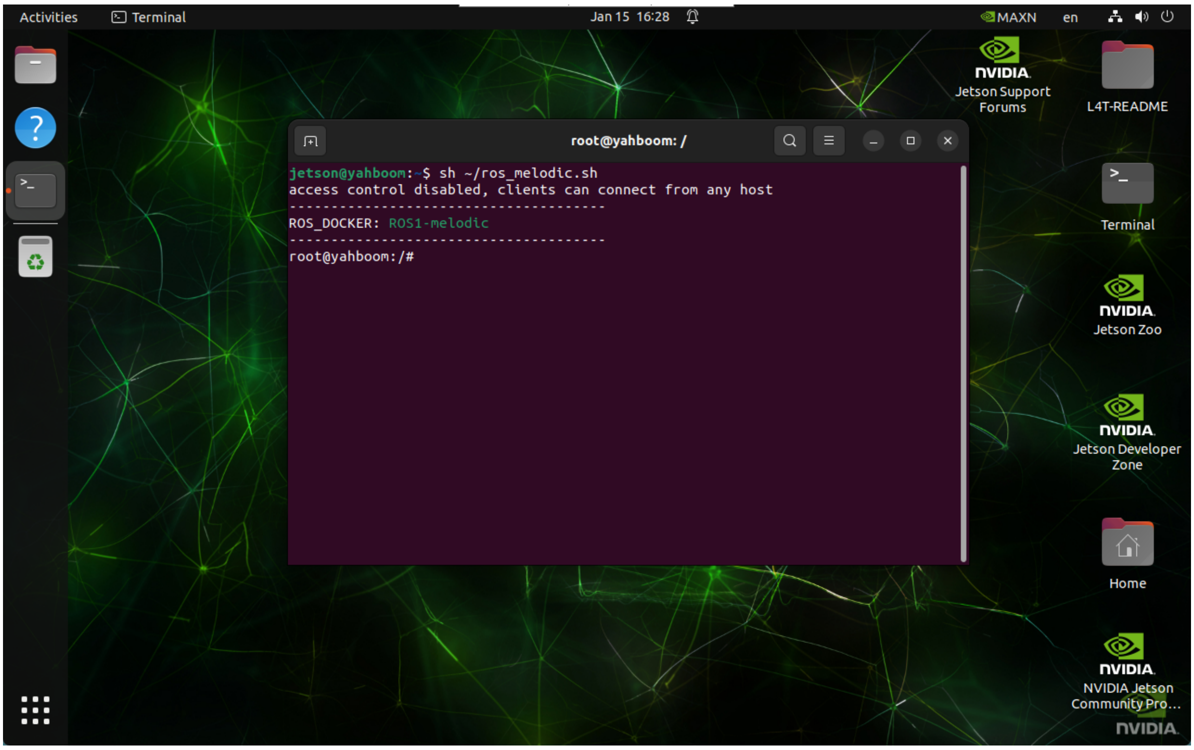
Opencv application

1. Enter Docker

```
sh ~/ros_melodic.sh
```



```
jetson@yahboom:~$ sh ~/ros_melodic.sh
access control disabled, clients can connect from any host
------------------------------------
ROS_DOCKER: ROS1-melodic
------------------------------------
root@yahboom:/#
```

2. Start the camera

Source code location in Docker: ~/yahboomcar_ws/src/opencv_apps/launch

```
roslaunch yahboomcar_visual opencv_apps.launch img_flip:=false
```

`img_flip`: Whether the image needs to be flipped horizontally, the default is false

3. Start the Opencv_apps function

Only one function can be run at a time, and the debug_view of some functions is closed. If there is no screen, you can view the effect in the following two ways.

```
The reason for closing debug_view is that an error will be generated in the
terminal, and the actual effect has no effect!
```

You need to start the camera and run this command:

```
roslaunch opencv_apps face_recognition.launch             # Face Recognition
roslaunch opencv_apps corner_harris.launch                # Harris corner
detection
roslaunch opencv_apps camshift.launch                     # Target Tracking
Algorithm
roslaunch opencv_apps contour_moments.launch              # Contour moment
roslaunch opencv_apps convex_hull.launch                  # Polygonal outline
roslaunch opencv_apps discrete_fourier_transform.launch   # Discrete Fourier
Transform Algorithm
roslaunch opencv_apps edge_detection.launch               # Edge detection
algorithm
roslaunch opencv_apps face_detection.launch               # Face Detection
Algorithm
roslaunch opencv_apps fback_flow.launch                   # Optical flow
detection algorithm
roslaunch opencv_apps find_contours.launch                # Contour Detection
roslaunch opencv_apps general_contours.launch             # General contour
detection
roslaunch opencv_apps goodfeature_track.launch            # Feature point
tracking
roslaunch opencv_apps hls_color_filter.launch             # HLS Color Filtering
roslaunch opencv_apps hough_circles.launch                # Hough circle
detection
roslaunch opencv_apps hough_lines.launch                  # Hough Line Detection
roslaunch opencv_apps hsv_color_filter.launch             # HSV color filtering
roslaunch opencv_apps lk_flow.launch                      # LK optical flow
algorithm
roslaunch opencv_apps people_detect.launch                # Human Detection
Algorithm
roslaunch opencv_apps phase_corr.launch                   # Phase-correlated
shift detection
roslaunch opencv_apps pyramids.launch                     # Image pyramid
sampling algorithm
roslaunch opencv_apps rgb_color_filter.launch             # RGB color filtering
roslaunch opencv_apps segment_objects.launch              # Clear background
detection algorithm
roslaunch opencv_apps simple_flow.launch                  # Streamlined optical
flow algorithm
roslaunch opencv_apps smoothing.launch                    # Simple filter
roslaunch opencv_apps threshold.launch                    # Threshold Image
Processing
roslaunch opencv_apps watershed_segmentation.launch       # Watershed
segmentation algorithm
```

4. Preview the screen

Start the camera and start one of the Opencv_apps functions to view the screen in the following way.

4.1. Local View

Enter the following command and select the corresponding topic to see the effect:

```
rqt_image_view
```

4.2. LAN View

In the same LAN, enter IP:port (8080) in the browser, for example:

```
192.168.2.150:8080 #IP refers to the host IP
```

References

wiki: http://wiki.ros.org/opencv_apps

Source code: https://github.com/ros-perception/opencv_apps.git

Most of the code was originally taken from https://github.com/Itseez/opencv/tree/master/samples/cpp