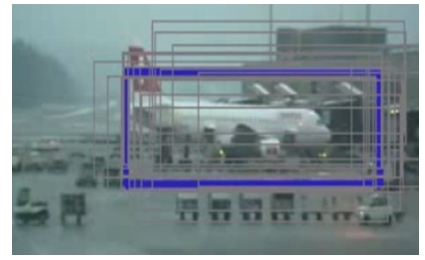
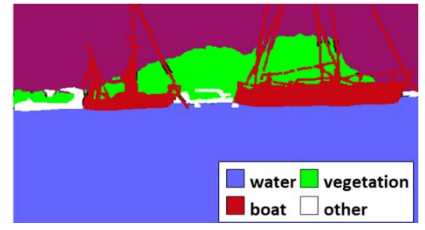








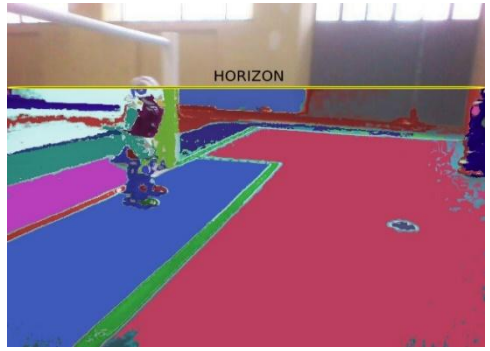
UNIVERSITÀ DEGLI STUDI DELLA BASILICATA

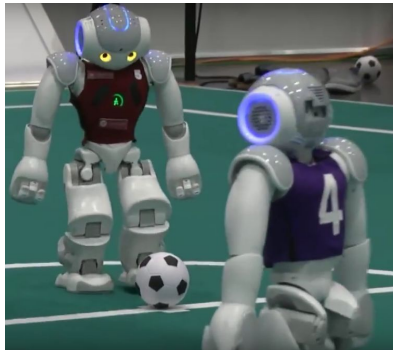
Corso di Sistemi Informativi
A.A. 2018/19

Docente
Domenico Daniele Bloisi

face detection

Maggio 2019




obiettivo

- Vogliamo realizzare un package ROS per la detection di volti
- Il package dovrà contenere due nodi:
  - il primo nodo servirà per rilevare i volti presenti nelle immagini a provenienti da una bag o da un sensore reale
  - il secondo nodo si occuperà di mostrare i risultati a video

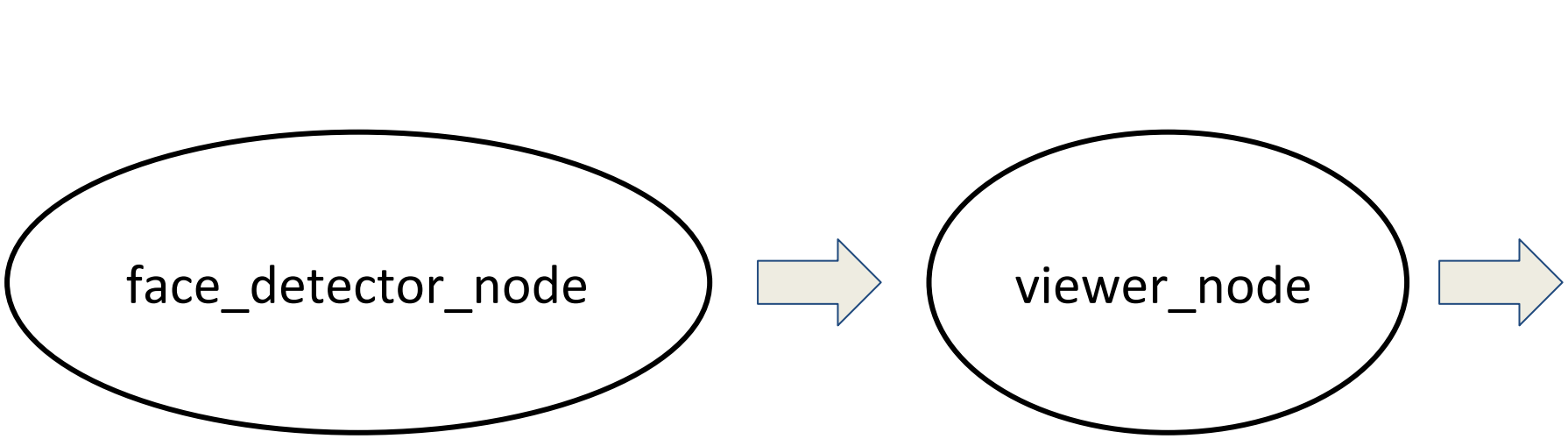
Package unibas_face_detector

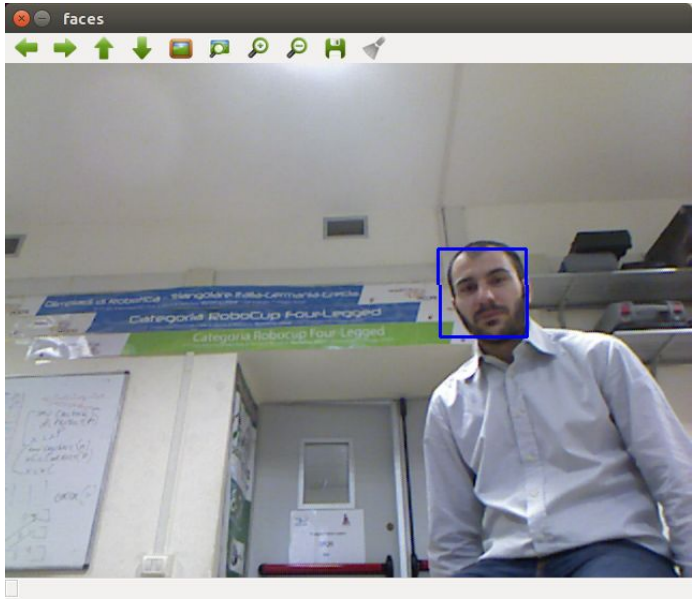
rosbag acquisita con una telecamera

face_detector_node

http://wiki.ros.org/cv_bridge/Tutorials/ConvertingBetweenROSImagesAndOpenCVImagesPython

face_detector_node e viewer_node





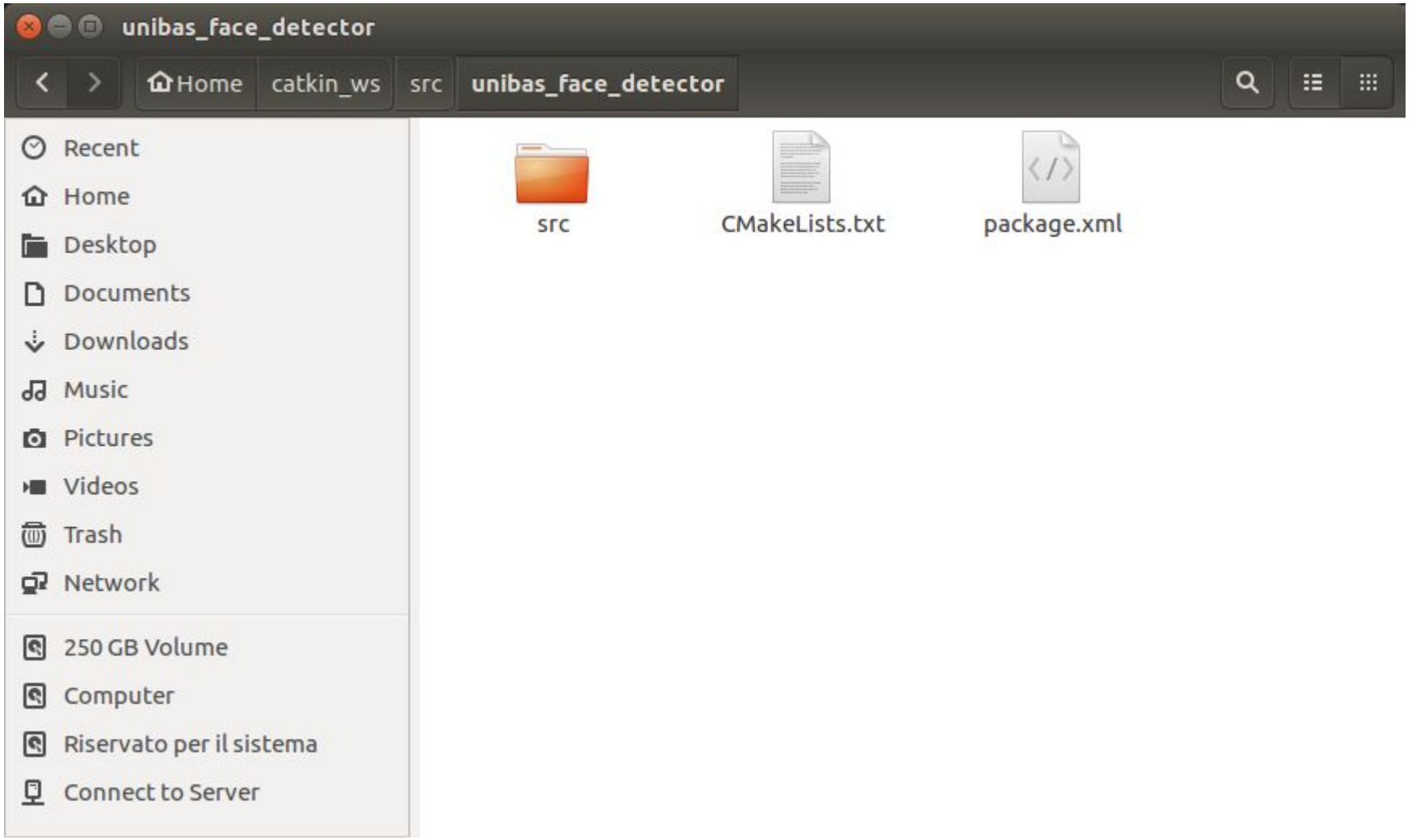
visualizzazione
immagine OpenCV

creazione unibas_face_detector

```
bloisi@bloisi-U36SG: ~/catkin_ws/src
bloisi@bloisi-U36SG:~$ cd ~/catkin_ws/src/
bloisi@bloisi-U36SG:~/catkin_ws/src$ catkin_create_pkg unibas_face_detector rospy
 std_msgs sensor_msgs cv_bridge
Created file unibas_face_detector/package.xml
Created file unibas_face_detector/CMakeLists.txt
Created folder unibas_face_detector/src
Successfully created files in /home/bloisi/catkin_ws/src/unibas_face_detector. Pl
ease adjust the values in package.xml.
bloisi@bloisi-U36SG:~/catkin_ws/src$
```

http://wiki.ros.org/cv_bridge/Tutorials/ConvertingBetweenROSImagesAndOpenCVImagesPython

cartella unibas_face_detector

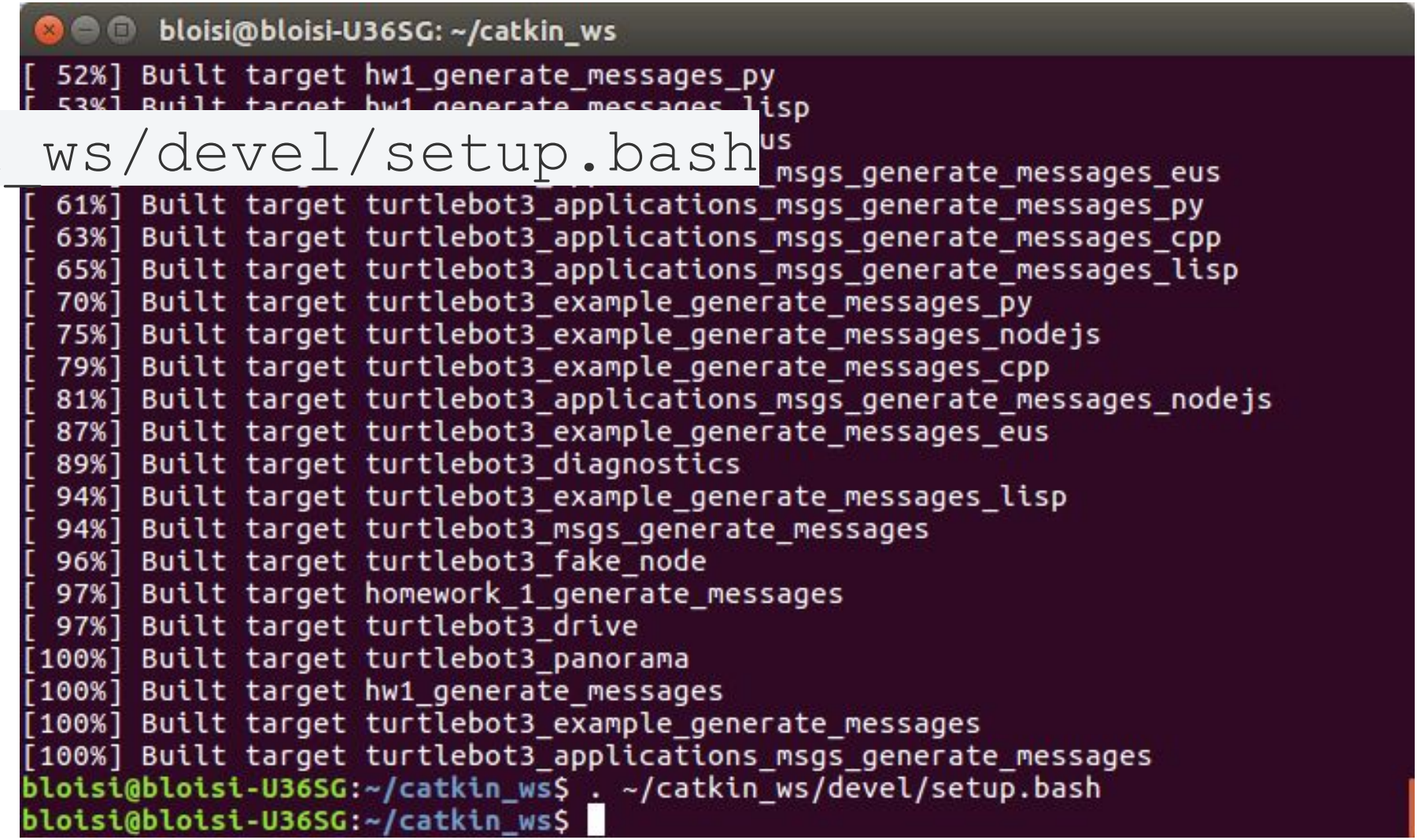


catkin_make

```
bloisi@bloisi-U36SG: ~/catkin_ws
bloisi@bloisi-U36SG:~$ cd ~/catkin_ws/src/
bloisi@bloisi-U36SG:~/catkin_ws/src$ catkin_create_pkg unibas_face_detector rospy
 std_msgs sensor_msgs cv_bridge
Created file unibas_face_detector/package.xml
Created file unibas_face_detector/CMakeLists.txt
Created folder unibas_face_detector/src
Successfully created files in /home/bloisi/catkin_ws/src/unibas_face_detector. Pl
ease adjust the values in package.xml.
bloisi@bloisi-U36SG:~/catkin_ws/src$ cd ..
bloisi@bloisi-U36SG:~/catkin_ws$ catkin_make
Base path: /home/bloisi/catkin_ws
Source space: /home/bloisi/catkin_ws/src
Build space: /home/bloisi/catkin_ws/build
Devel space: /home/bloisi/catkin_ws/devel
Install space: /home/bloisi/catkin_ws/install
####
#### Running command: "cmake /home/bloisi/catkin_ws/src -DCATKIN_DEVEL_PREFIX=/ho
me/bloisi/catkin_ws/devel -DCMAKE_INSTALL_PREFIX=/home/bloisi/catkin_ws/install -
G Unix Makefiles" in "/home/bloisi/catkin_ws/build"
####
-- Using CATKIN_DEVEL_PREFIX: /home/bloisi/catkin_ws/devel
-- Using CMAKE_PREFIX_PATH: /home/bloisi/catkin_ws/devel;/opt/ros/kinetic
-- This workspace overlays: /home/bloisi/catkin_ws/devel;/opt/ros/kinetic
-- Using PYTHON_EXECUTABLE: /usr/bin/python
```

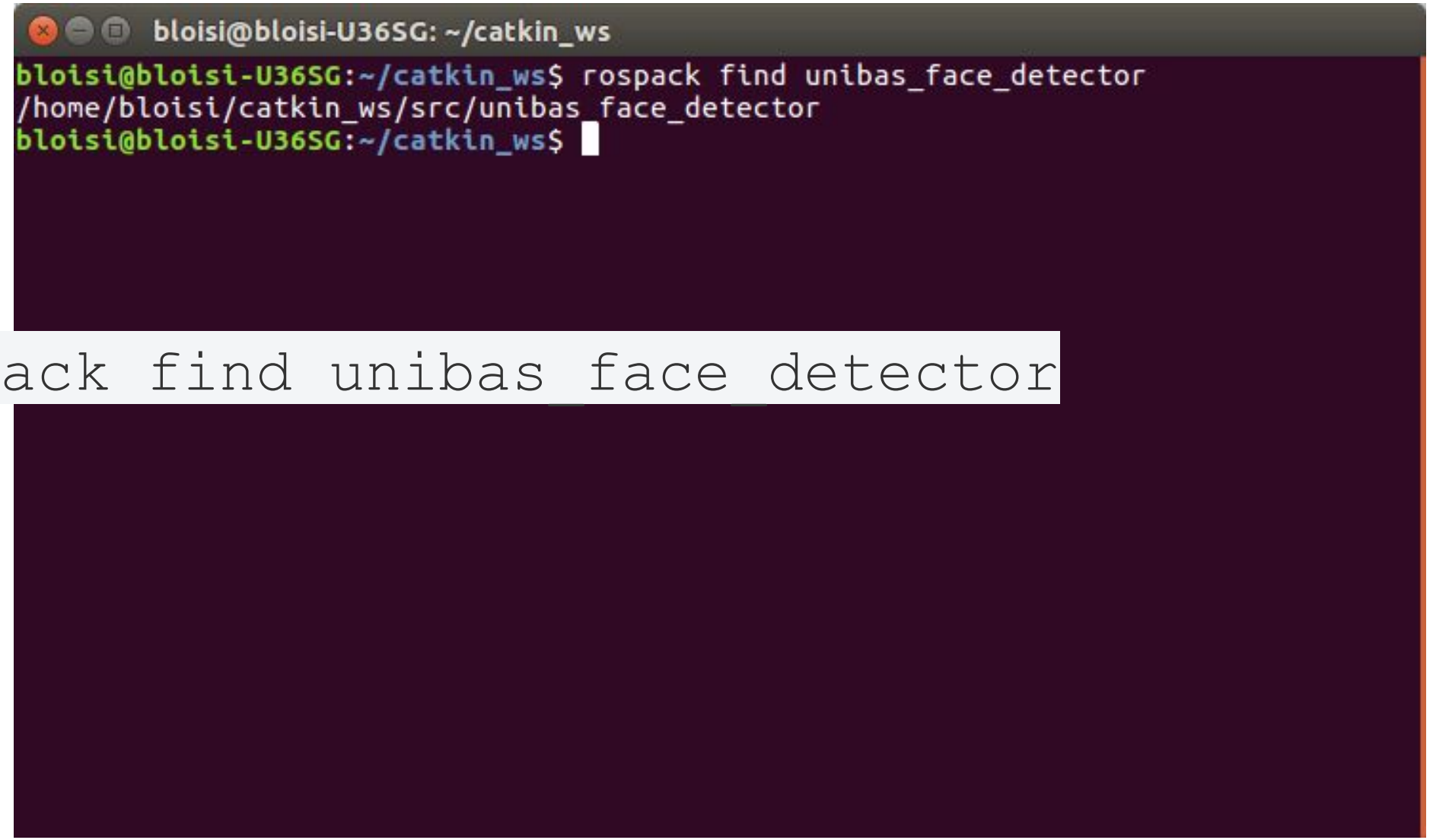
settiamo l’ambiente ROS

```
. ~/catkin_ws/devel/setup.bash
```



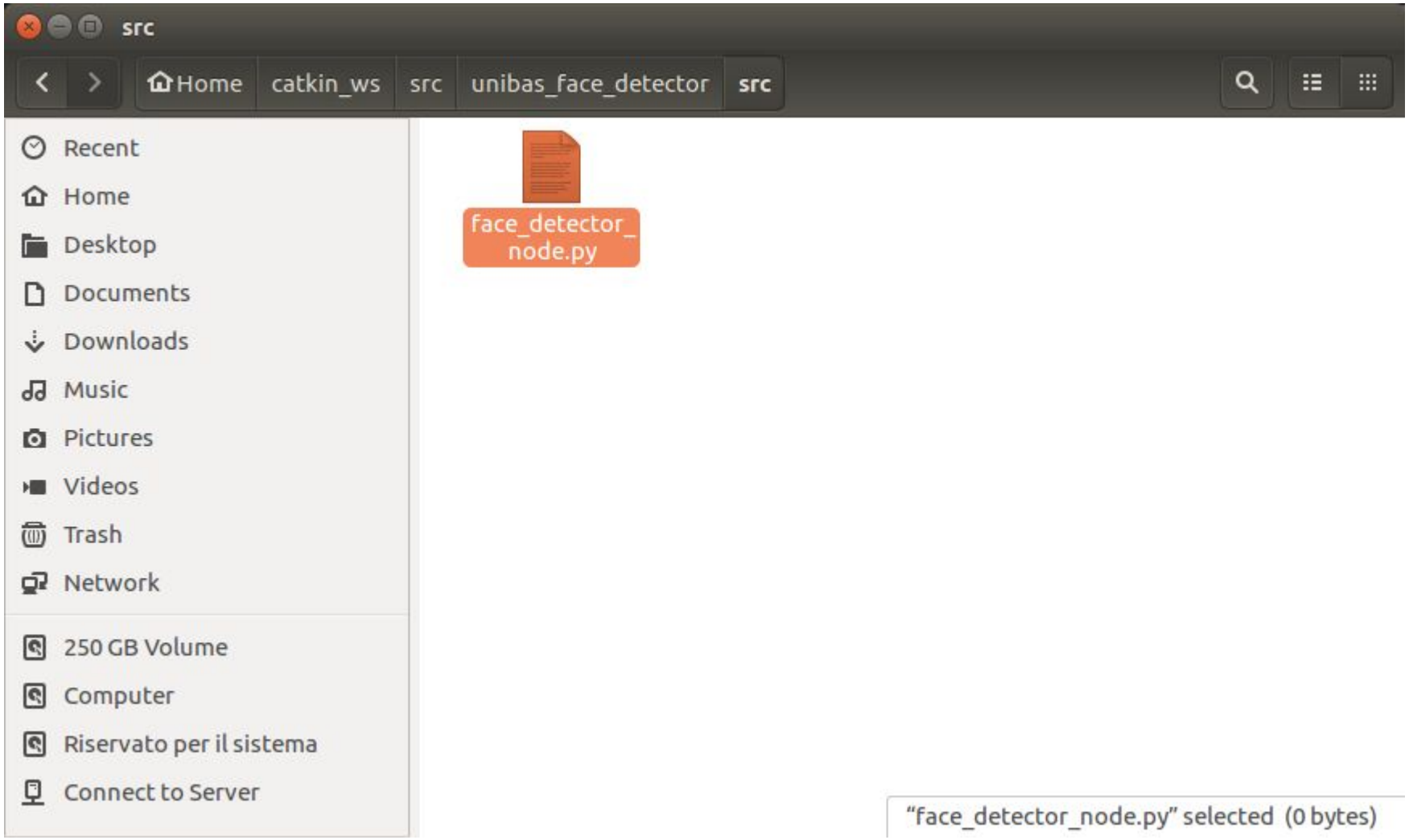
https://answers.ros.org/question/229365/do-i-really-need-to-source-catkin_wsdevelsetupbash/

rospack find




```
rospack find unibas_face_detector
```

creiamo face_detector_node.py



codice face_detector_node.py

face_detector_node.py (~/catkin_ws/src/unibas_face_detector/src) - gedit

Open Save

```
1 #!/usr/bin/env python
2 from __future__ import print_function
3
4 import roslib
5 roslib.load_manifest('unibas_face_detector')
6 import sys
7 import rospy
8 import cv2
9 import numpy as np
10 import message_filters
11 from std_msgs.msg import String
12 from sensor_msgs.msg import Image
13 from cv_bridge import CvBridge, CvBridgeError
14
15 class face_detector:
16
17   def __init__(self):
18     self.bridge = CvBridge()
19
20     self.image_sub = rospy.Subscriber("/camera/rgb/image_raw", Image, self.callback)
21
22     self.pub = rospy.Publisher('/unibas_face_detector/faces', Image, queue_size=1)
23
```

Python Tab Width: 8 Ln 48, Col 21 INS

codice face_detector_node.py

face_detector_node.py (~/catkin_ws/src/unibas_face_detector/src) - gedit

Open Save

```
24  def callback(self, rgb_data):
25
26    try:
27      img = self.bridge.imgmsg_to_cv2(rgb_data, "bgr8")
28      face_cascade = cv2.CascadeClassifier('/opt/ros/kinetic/share/OpenCV-3.3.1-dev/haarcascades/
  haarcascade_frontalface_default.xml')
29      gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
30      faces = face_cascade.detectMultiScale(gray, 1.3, 5)
31      for (x,y,w,h) in faces:
32        cv2.rectangle(img,(x,y),(x+w,y+h),(255,0,0),2)
33        roi_gray = gray[y:y+h, x:x+w]
34        roi_color = img[y:y+h, x:x+w]
35
36    except CvBridgeError as e:
37      print(e)
38
39    #convert opencv format back to ros format and publish result
40    try:
41      faces_message = self.bridge.cv2_to_imgmsg(img, "bgr8")
42      self.pub.publish(faces_message)
43    except CvBridgeError as e:
44      print(e)
45
```

Python Tab Width: 8 Ln 48, Col 21 INS

codice face_detector_node.py

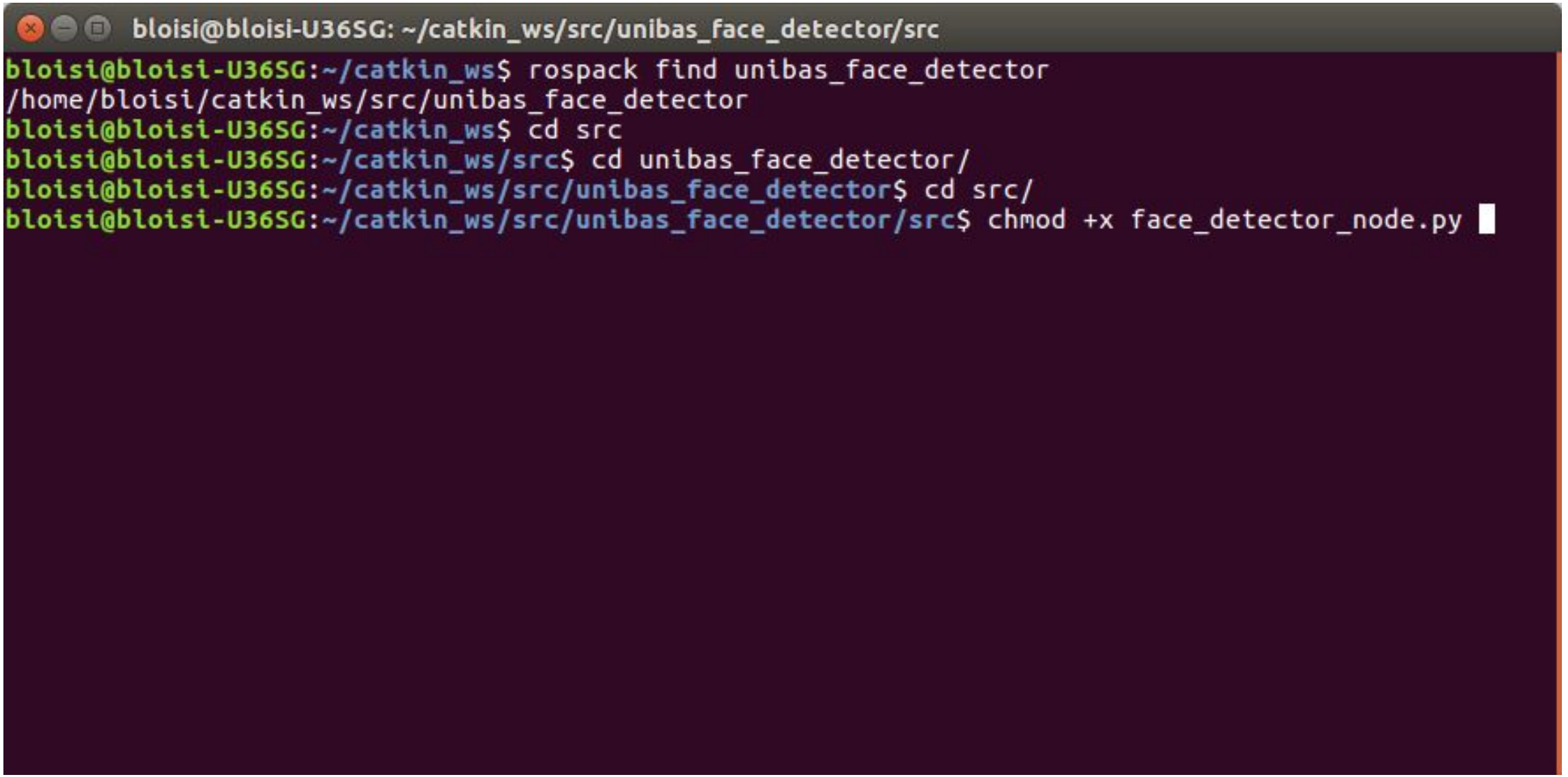
face_detector_node.py (~/catkin_ws/src/unibas_face_detector/src) - gedit

Open Save

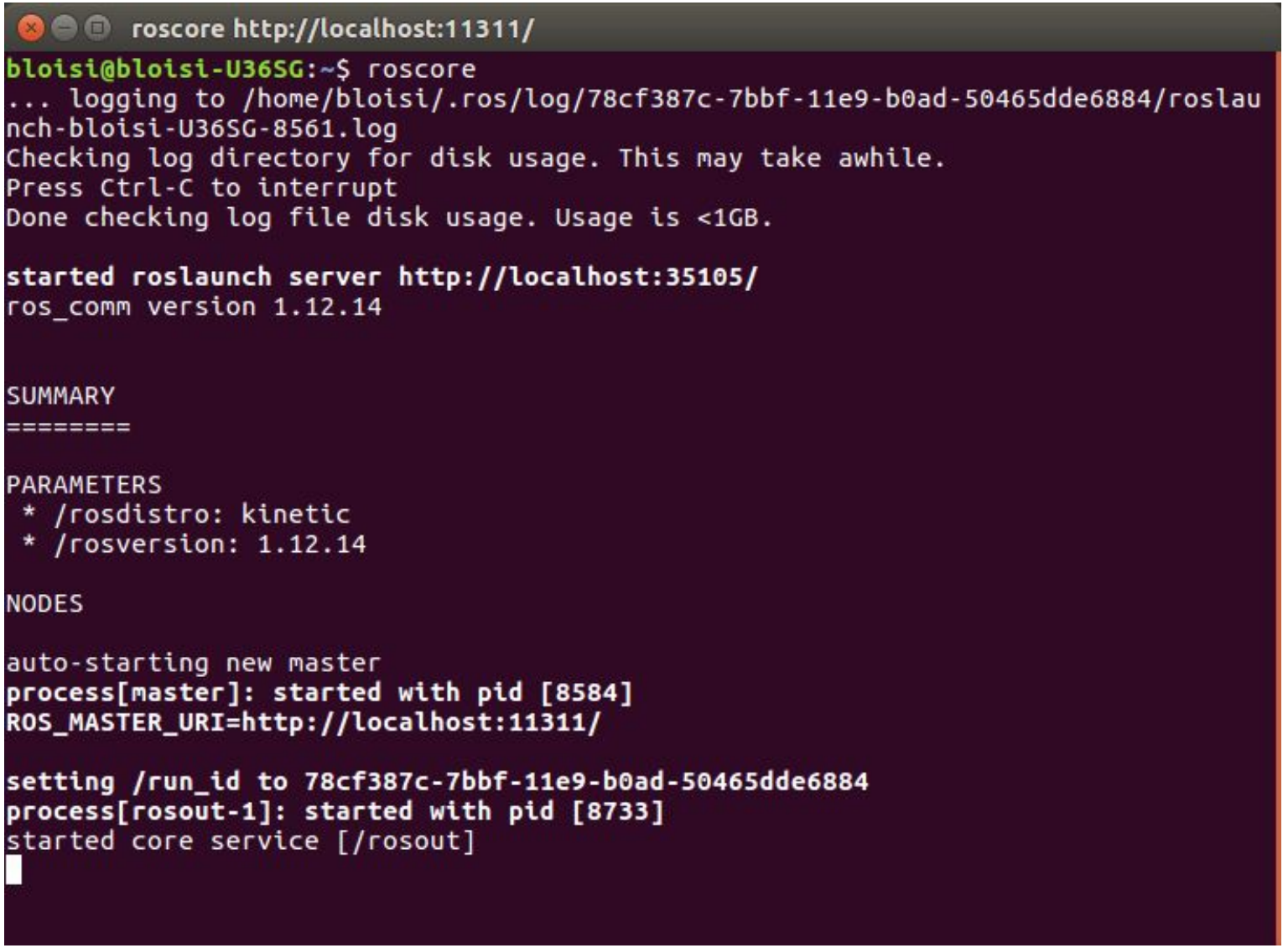
```
46
47 def main(args):
48   fd = face_detector()
49   rospy.init_node('face_detector_node', anonymous=True)
50   try:
51     rospy.spin()
52   except KeyboardInterrupt:
53     print("Shutting down")
54
55 if __name__ == '__main__':
56     main(sys.argv)
57
```

Python Tab Width: 8 Ln 48, Col 21 INS

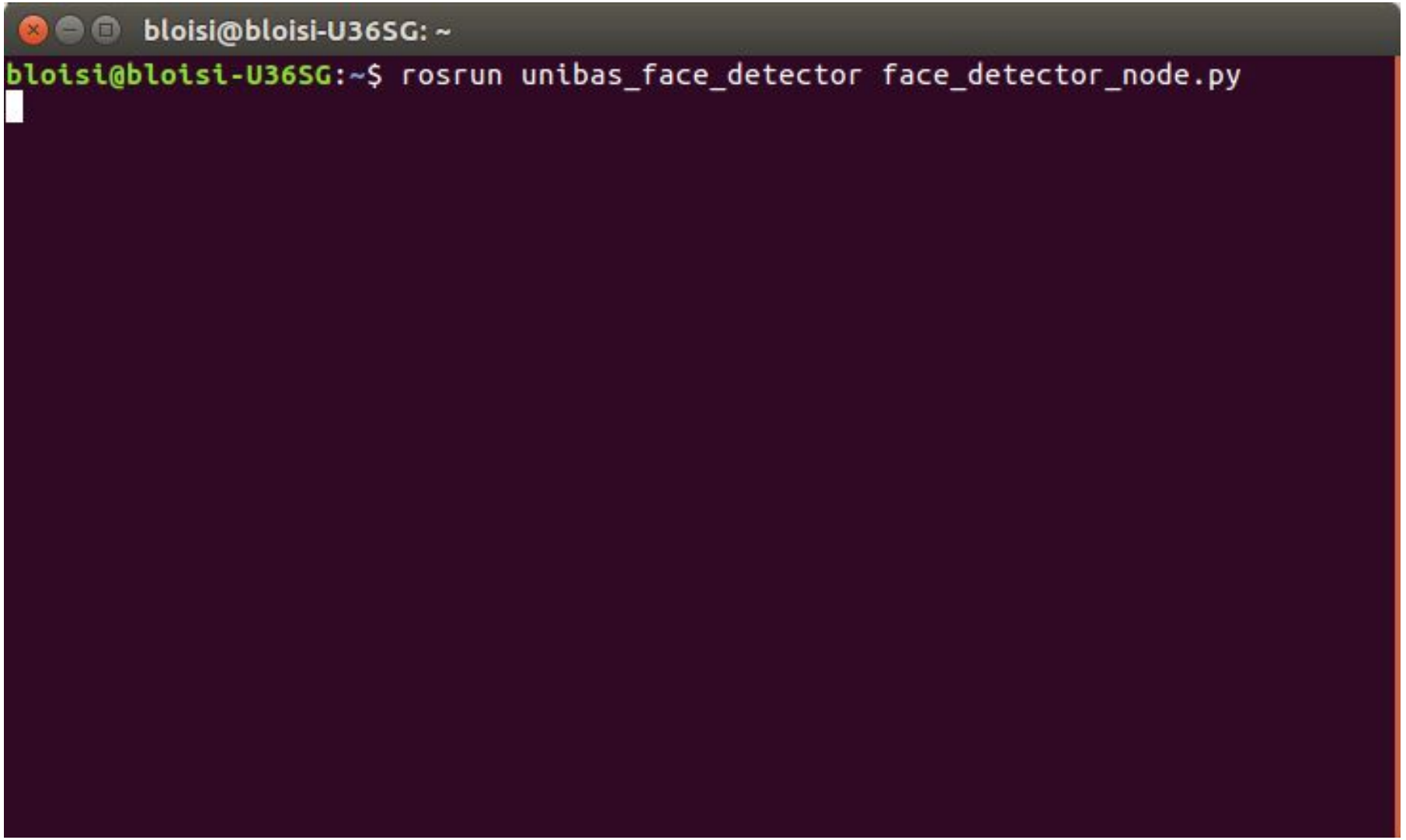
permessi per face_detector_node.py



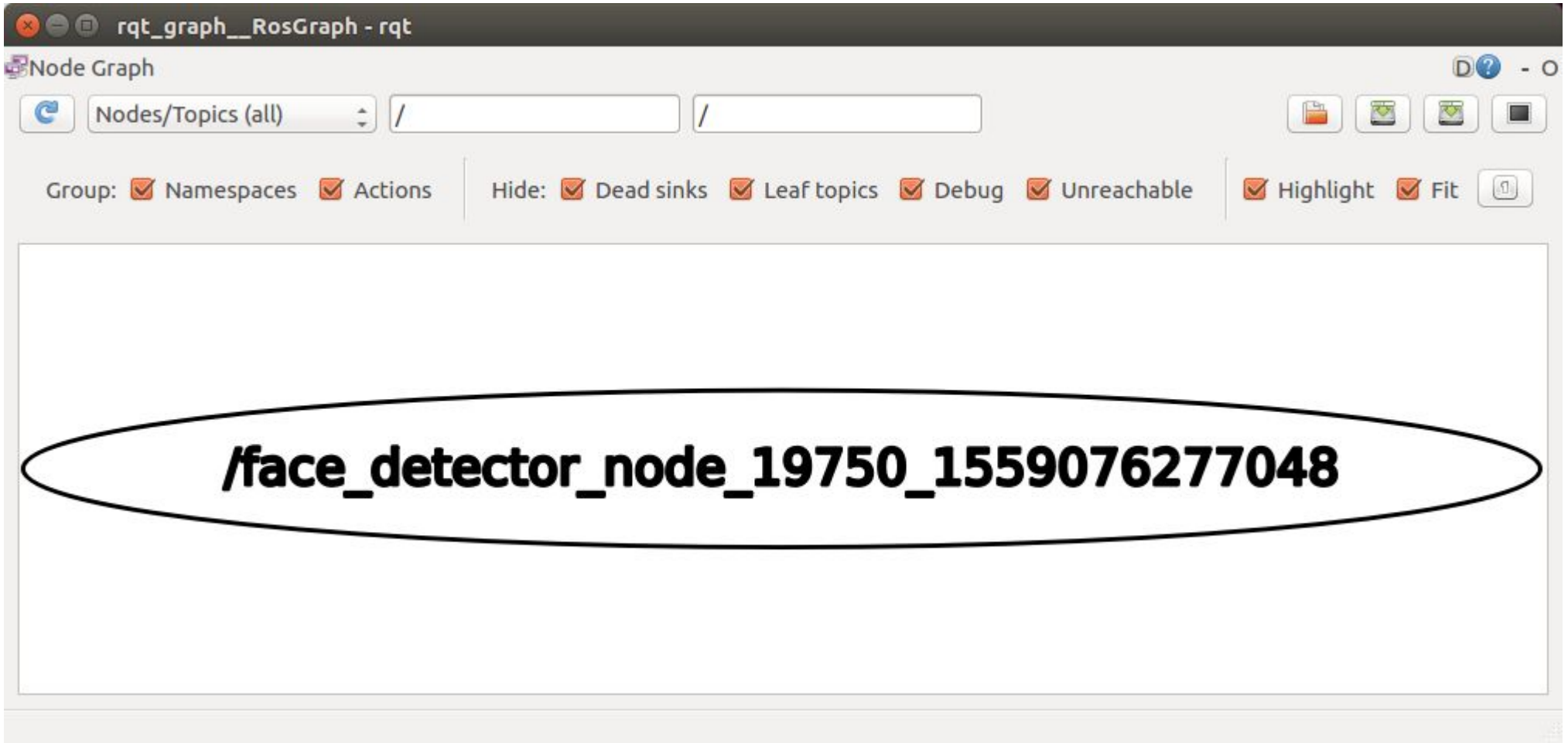
roscore


roscore http://localhost:11311/
bloisi@bloisi-U36SG:~$ roscore
... logging to /home/bloisi/.ros/log/78cf387c-7bbf-11e9-b0ad-50465dde6884/roslaunch-bloisi-U36SG-8561.log
Checking log directory for disk usage. This may take awhile.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.
started roslaunch server http://localhost:35105/
ros_comm version 1.12.14
SUMMARY
========
PARAMETERS
* /rosdistro: kinetic
* /rosversion: 1.12.14
NODES
auto-starting new master
process[master]: started with pid [8584]
ROS_MASTER_URI=http://localhost:11311/
setting /run_id to 78cf387c-7bbf-11e9-b0ad-50465dde6884
process[rosout-1]: started with pid [8733]
started core service [/rosout]

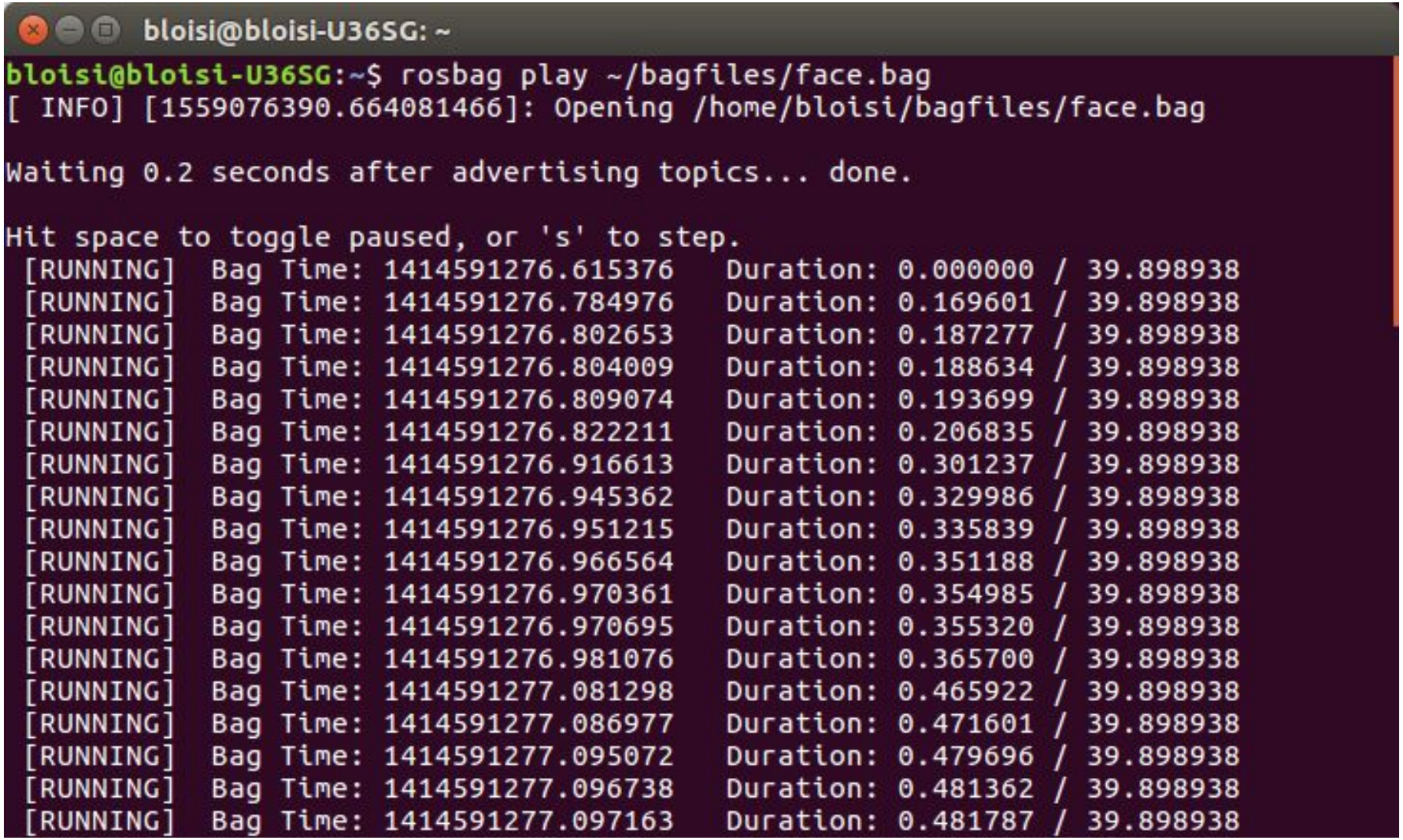
rosrun face_detector_node



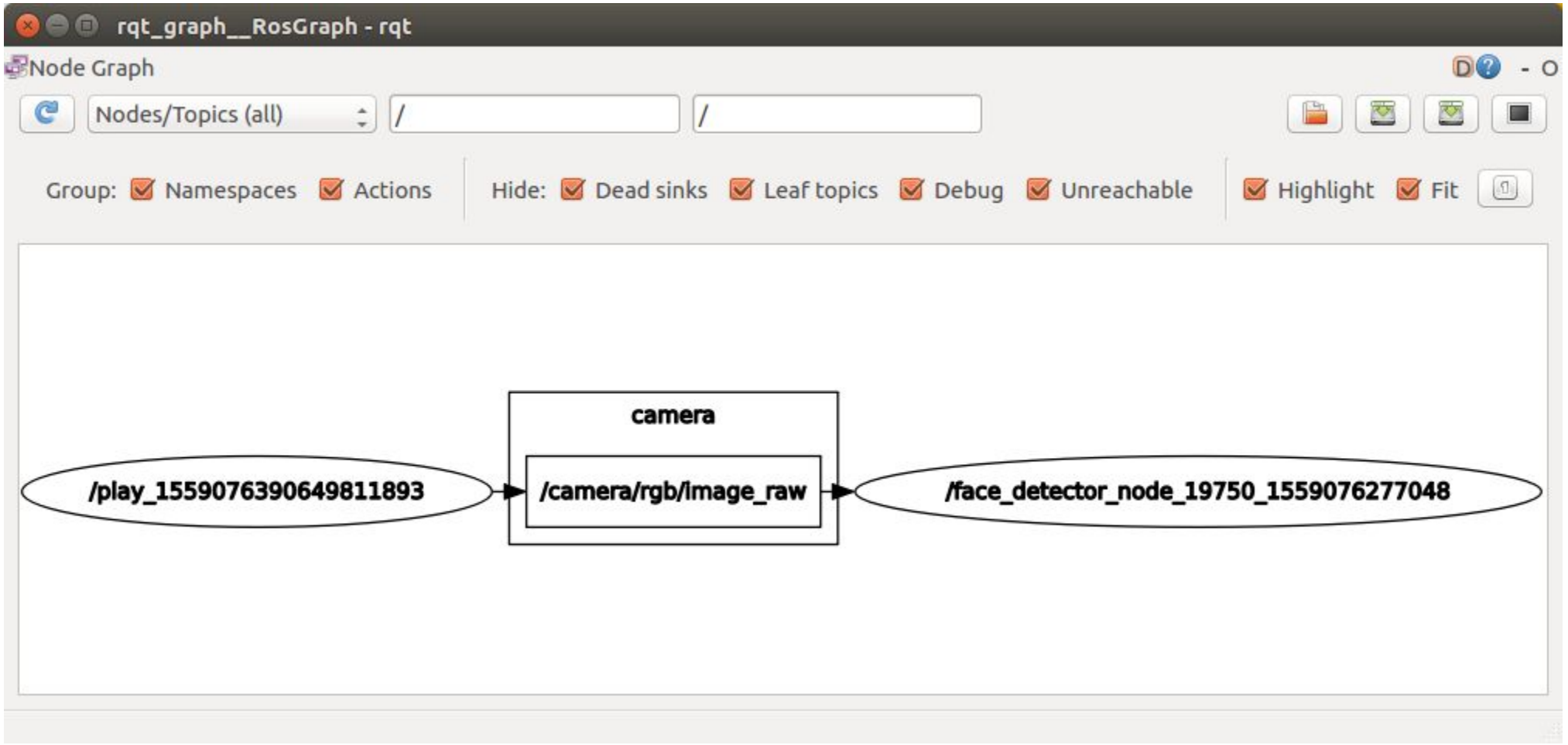
rqt_graph


rqt_graph__RosGraph - rqt
Node Graph
Nodes/Topics (all)
/
/
Group: Namespaces Actions
Hide: Dead sinks Leaf topics Debug Unreachable
Highlight Fit
/face_detector_node_19750_1559076277048

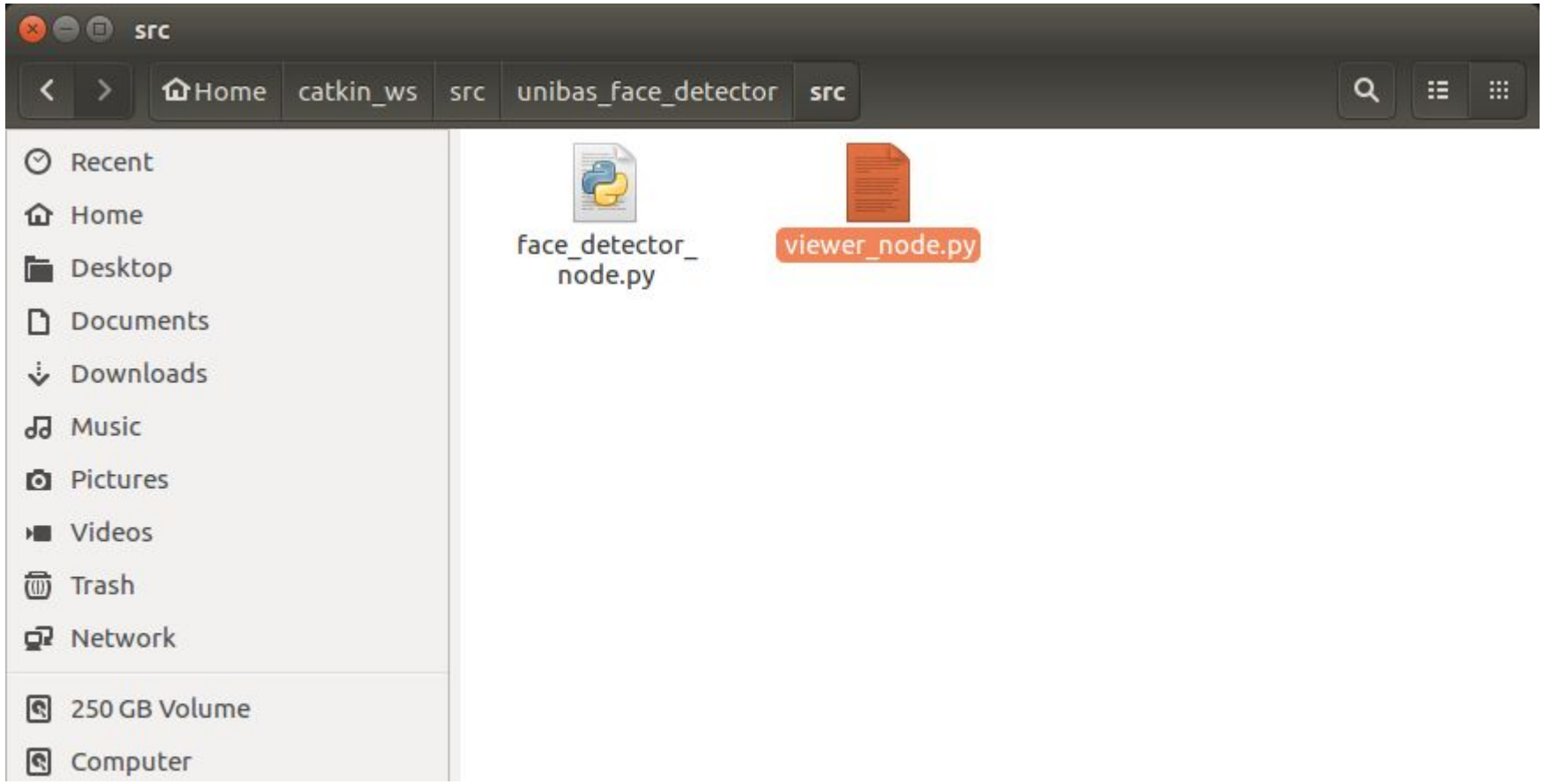
rosbag play


bloisi@bloisi-U36SG: ~
bloisi@bloisi-U36SG:~$ rosbag play ~/bagfiles/face.bag
[INFO] [1559076390.664081466]: Opening /home/bloisi/bagfiles/face.bag
Waiting 0.2 seconds after advertising topics... done.
Hit space to toggle paused, or 's' to step.
[RUNNING] Bag Time: 1414591276.615376 Duration: 0.000000 / 39.898938
[RUNNING] Bag Time: 1414591276.784976 Duration: 0.169601 / 39.898938
[RUNNING] Bag Time: 1414591276.802653 Duration: 0.187277 / 39.898938
[RUNNING] Bag Time: 1414591276.804009 Duration: 0.188634 / 39.898938
[RUNNING] Bag Time: 1414591276.809074 Duration: 0.193699 / 39.898938
[RUNNING] Bag Time: 1414591276.822211 Duration: 0.206835 / 39.898938
[RUNNING] Bag Time: 1414591276.916613 Duration: 0.301237 / 39.898938
[RUNNING] Bag Time: 1414591276.945362 Duration: 0.329986 / 39.898938
[RUNNING] Bag Time: 1414591276.951215 Duration: 0.335839 / 39.898938
[RUNNING] Bag Time: 1414591276.966564 Duration: 0.351188 / 39.898938
[RUNNING] Bag Time: 1414591276.970361 Duration: 0.354985 / 39.898938
[RUNNING] Bag Time: 1414591276.970695 Duration: 0.355320 / 39.898938
[RUNNING] Bag Time: 1414591276.981076 Duration: 0.365700 / 39.898938
[RUNNING] Bag Time: 1414591277.081298 Duration: 0.465922 / 39.898938
[RUNNING] Bag Time: 1414591277.086977 Duration: 0.471601 / 39.898938
[RUNNING] Bag Time: 1414591277.095072 Duration: 0.479696 / 39.898938
[RUNNING] Bag Time: 1414591277.096738 Duration: 0.481362 / 39.898938
[RUNNING] Bag Time: 1414591277.097163 Duration: 0.481787 / 39.898938

rqt_graph


rqt_graph__RosGraph - rqt
Node Graph
Nodes/Topics (all)
/
/
Group: Namespaces Actions
Hide: Dead sinks Leaf topics Debug Unreachable
Highlight Fit
camera
/play_1559076390649811893
/camera/rgb/image_raw
/face_detector_node_19750_1559076277048

viewer_node


src
Home
catkin_ws
src
unibas_face_detector
src
Recent
Home
Desktop
Documents
Downloads
Music
Pictures
Videos
Trash
Network
250 GB Volume
Computer
face_detector_
node.py
viewer_node.py

codice viewer_node

viewer_node.py (~/catkin_ws/src/unibas_face_detector/src) - gedit

Open Save

```
1 #!/usr/bin/env python
2 from __future__ import print_function
3
4 import roslib
5 roslib.load_manifest('unibas_face_detector')
6 import sys
7 import rospy
8 import cv2
9 from std_msgs.msg import String
10 from sensor_msgs.msg import Image
11 from cv_bridge import CvBridge, CvBridgeError
12
13 class viewer:
14
15   def __init__(self):
16     self.bridge = CvBridge()
17     self.image_sub = rospy.Subscriber("/unibas_face_detector/faces", Image, self.callback)
18
19   def callback(self,data):
20     try:
21       cv_image = self.bridge.imgmsg_to_cv2(data, "bgr8")
22     except CvBridgeError as e:
23       print(e)
24
25     cv2.imshow("faces", cv_image)
26     cv2.waitKey(30)
27
```

Python Tab Width: 8 Ln 30, Col 3 INS

codice viewer_node

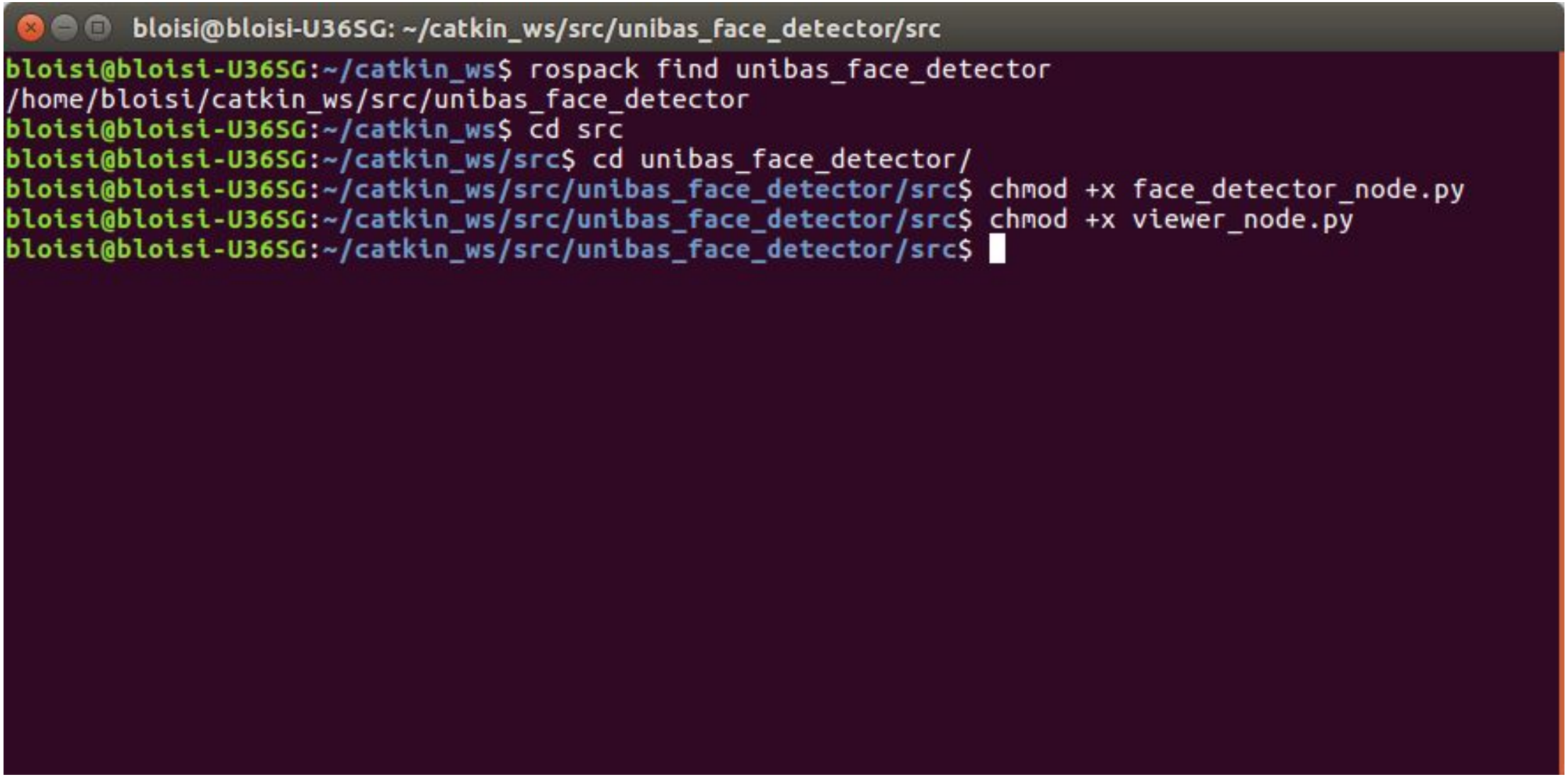
viewer_node.py (~/catkin_ws/src/unibas_face_detector/src) - gedit

Open Save

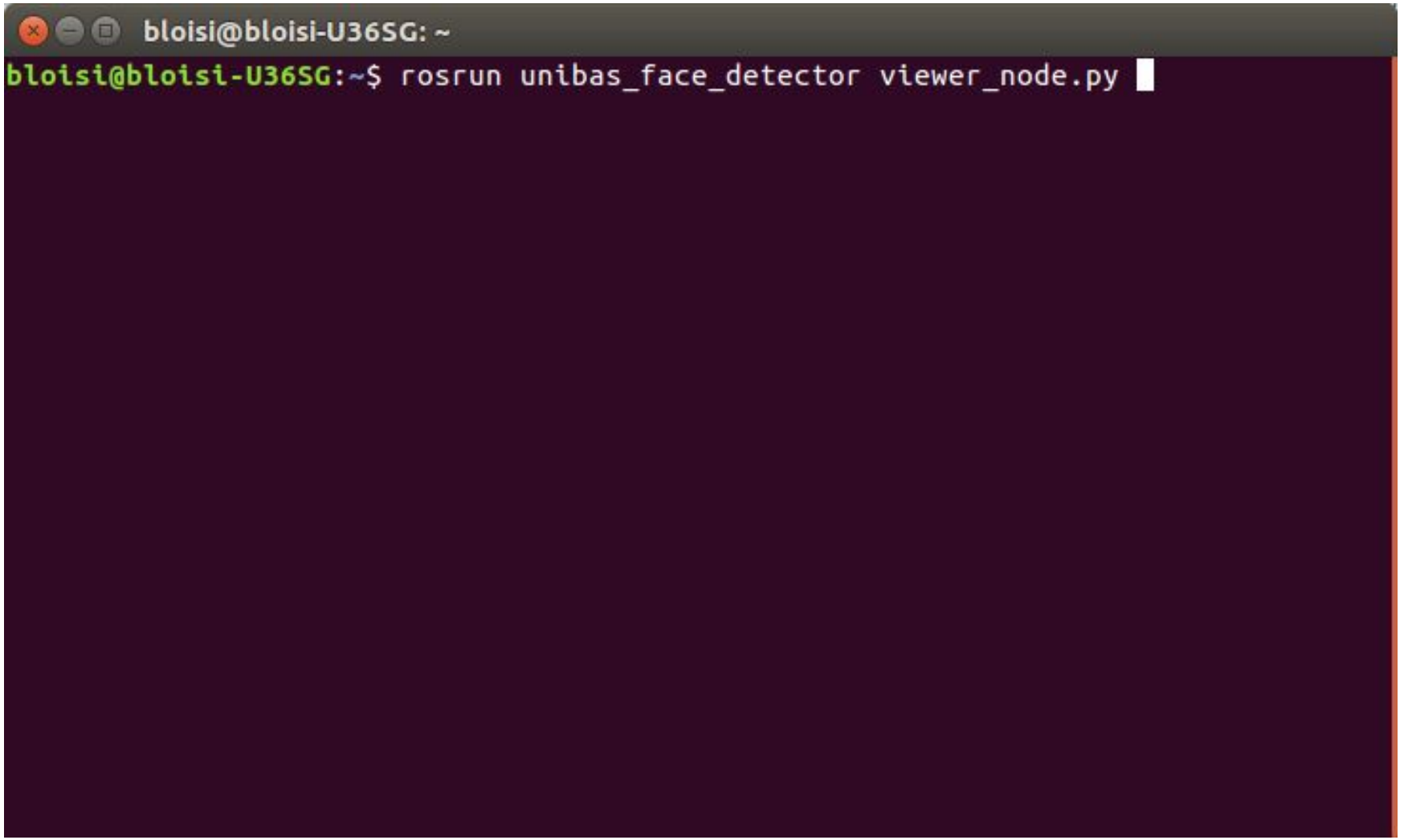
```
28
29 def main(args):
30   v = viewer()
31   rospy.init_node('viewer_node', anonymous=True)
32   try:
33     rospy.spin()
34   except KeyboardInterrupt:
35     print("Shutting down")
36   cv2.destroyAllWindows()
37
38 if __name__ == '__main__':
39     main(sys.argv)
40
```

Python Tab Width: 8 Ln 30, Col 3 INS

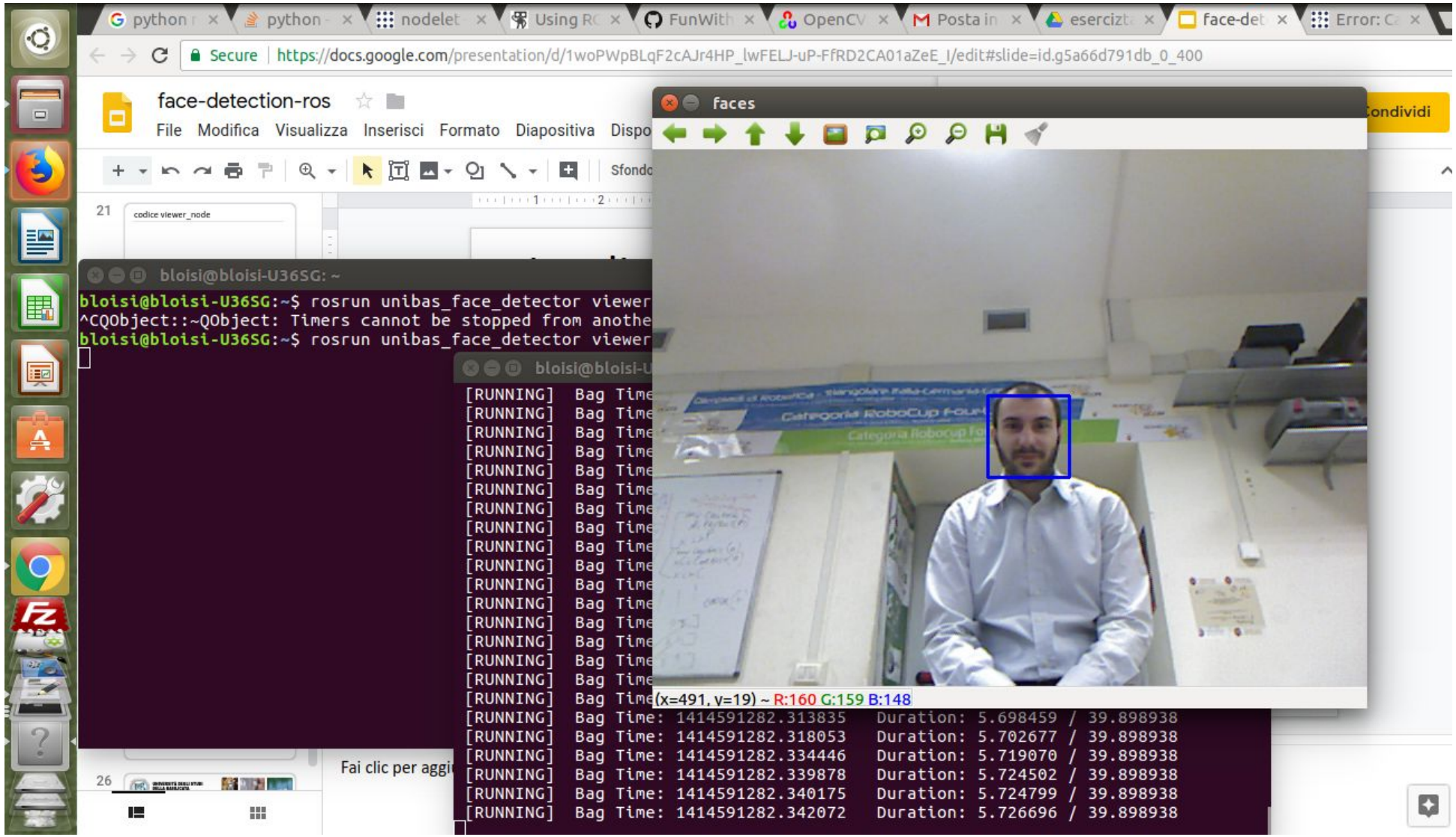
permessi per viewer_node.py



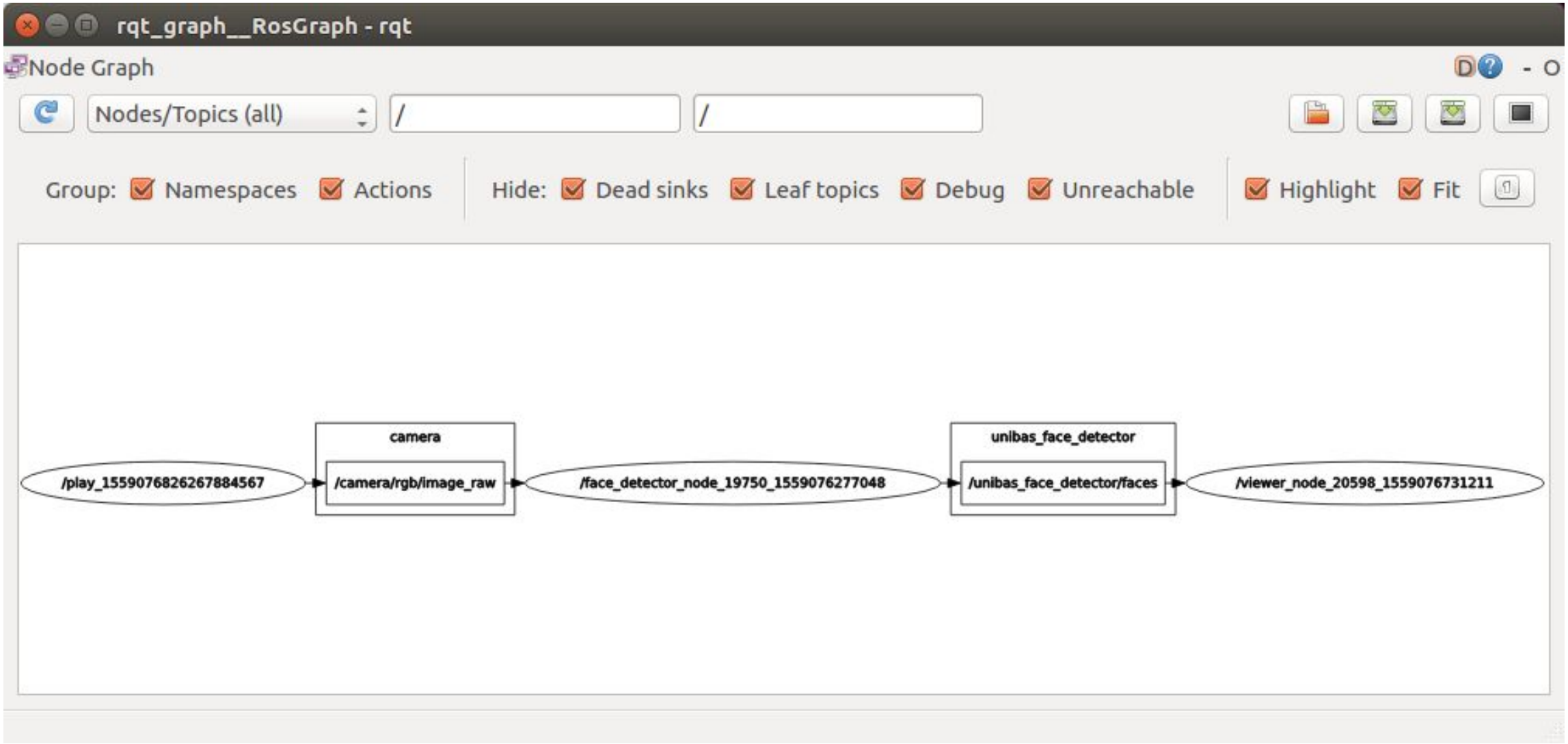
rosrun viewer_node.py



visualizzazione

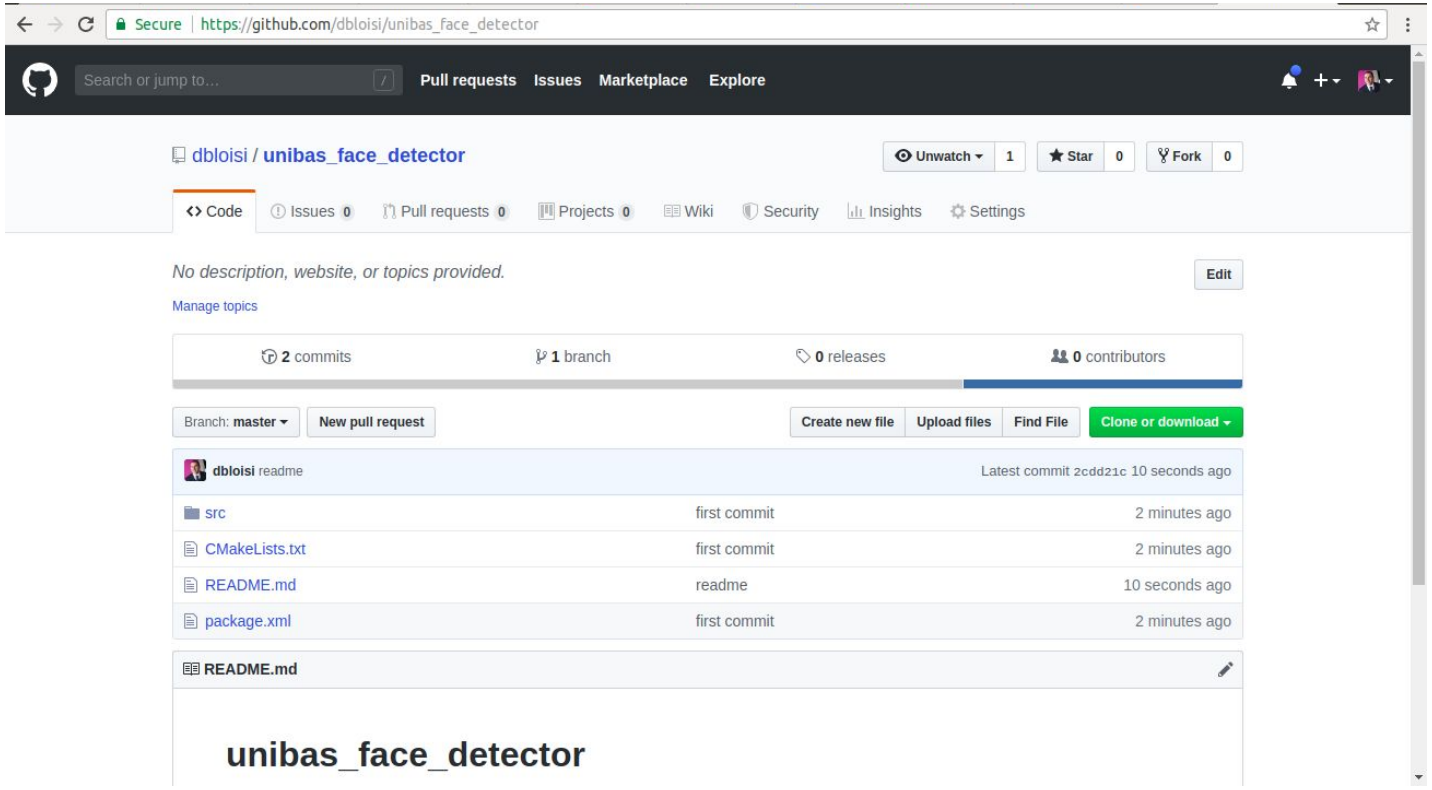


rqt_graph


rqt_graph__RosGraph - rqt
Node Graph
Nodes/Topics (all)
/
/
Group: Namespaces Actions
Hide: Dead sinks Leaf topics Debug Unreachable
Highlight Fit
camera
/play_1559076826267884567
/camera/rgb/image_raw
/face_detector_node_19750_1559076277048
unibas_face_detector
/unibas_face_detector/faces
/viewer_node_20598_1559076731211

repository unibas_face_detector

Il codice del repository unibas_face_detector è disponibile al seguente link https://github.com/dbloisi/unibas_face_detector

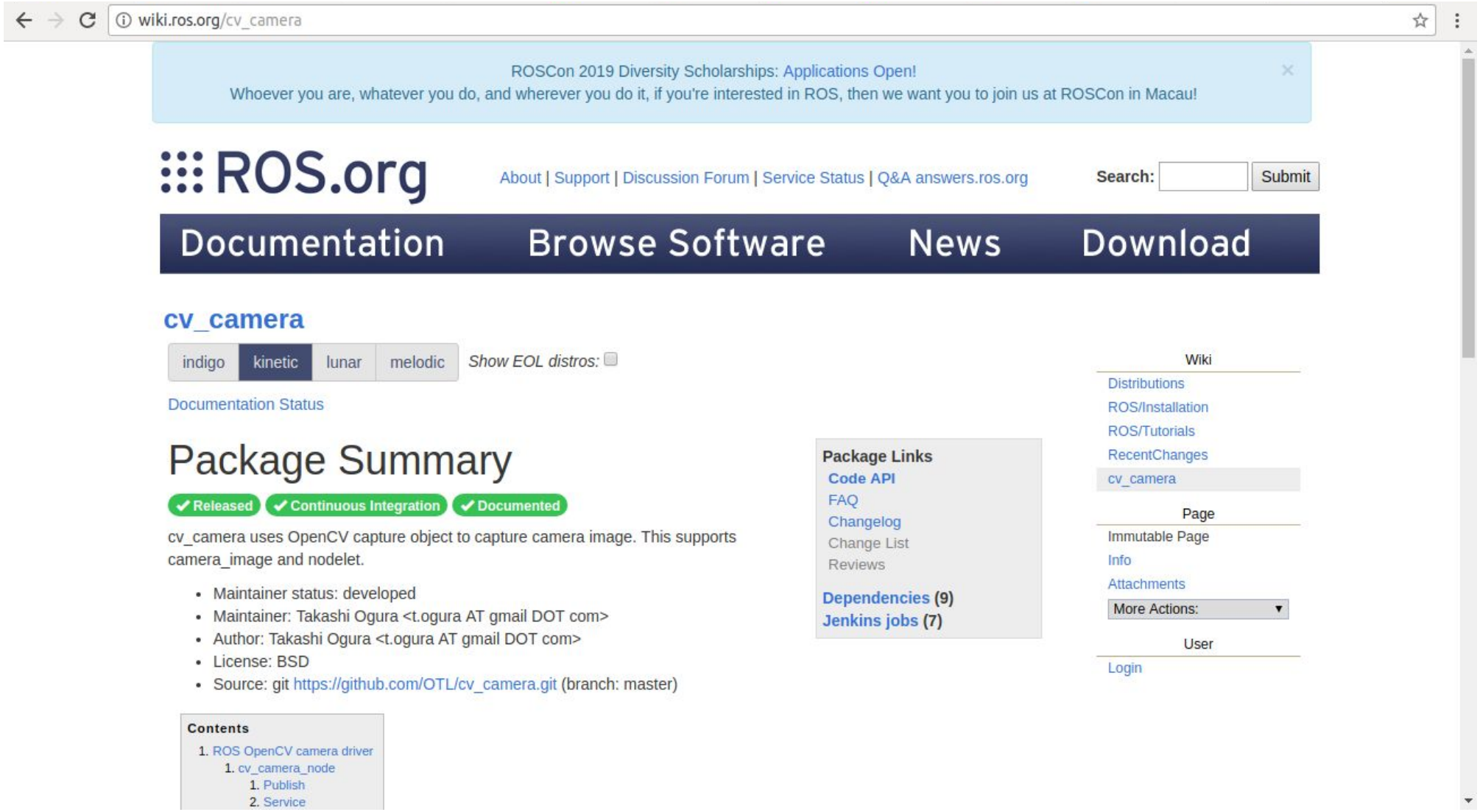


input da webcam

Proviamo ad utilizzare le immagini provenienti dalla webcam del nostro pc

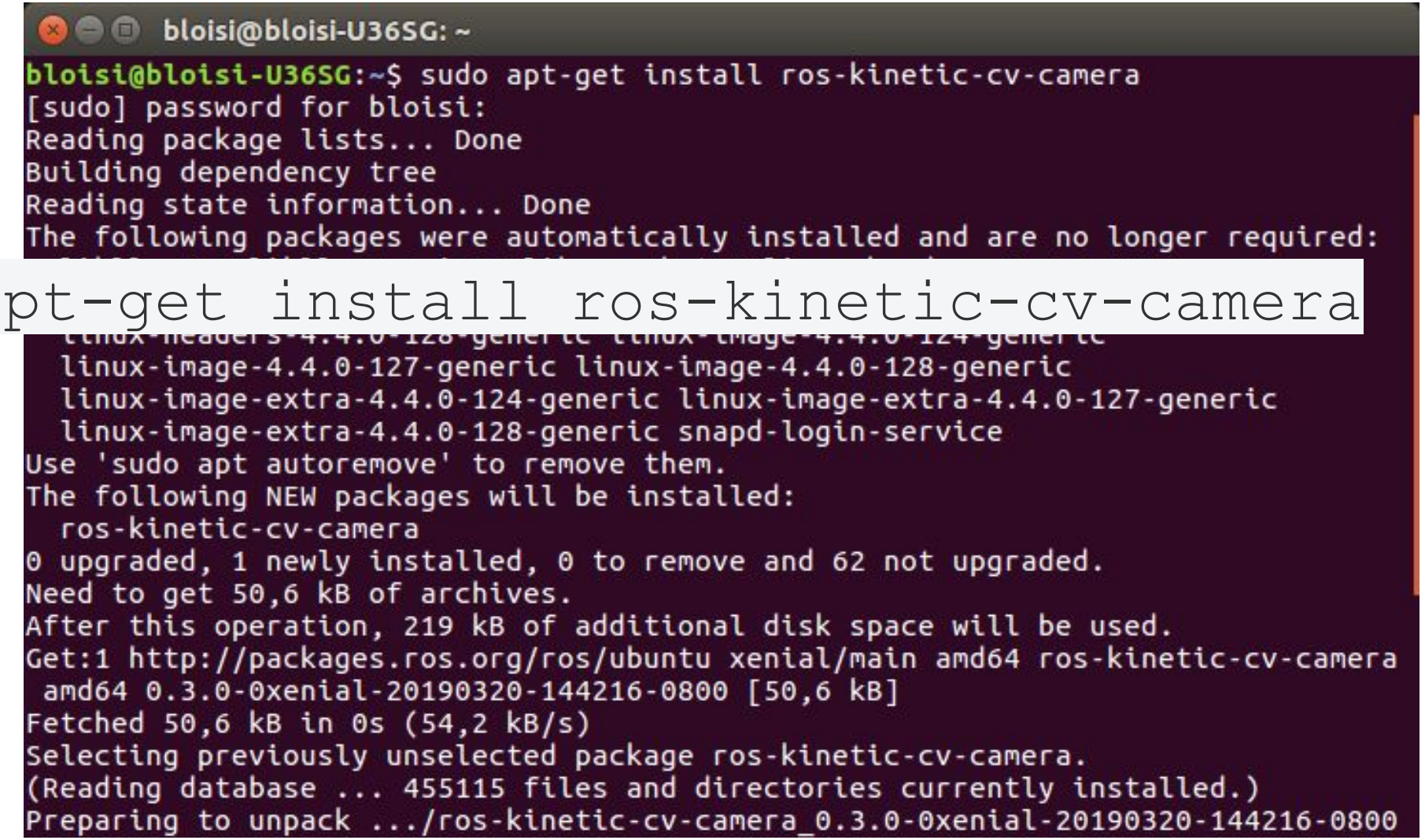
Ci servirà un nodo che legga lo stream della webcam e lo pubblichi su un topic ROS

package cv_camera



installazione package cv_camera

```
sudo apt-get install ros-kinetic-cv-camera
```



lanciamo cv_camera

wiki.ros.org/cv_camera

- ~image_raw (sensor_msgs/Image)
- ~camera_info (sensor_msgs/CameraInfo)

1.2 Service

- ~set_camera_info (sensor_msgs/SetCameraInfo)

1.3 Parameters

- ~rate (double: default 30.0) publish rate [Hz].
- ~device_id (int: default 0) capture device id.
- ~frame_id (string: default "camera") frame_id of message header.
- ~image_width (int) try to set capture image width.
- ~image_height (int) try to set capture image height.
- ~camera_info_url (string) url of camera info yaml.
- ~file (string: default "") if not "" then use movie file instead of device.

1.4 Usage

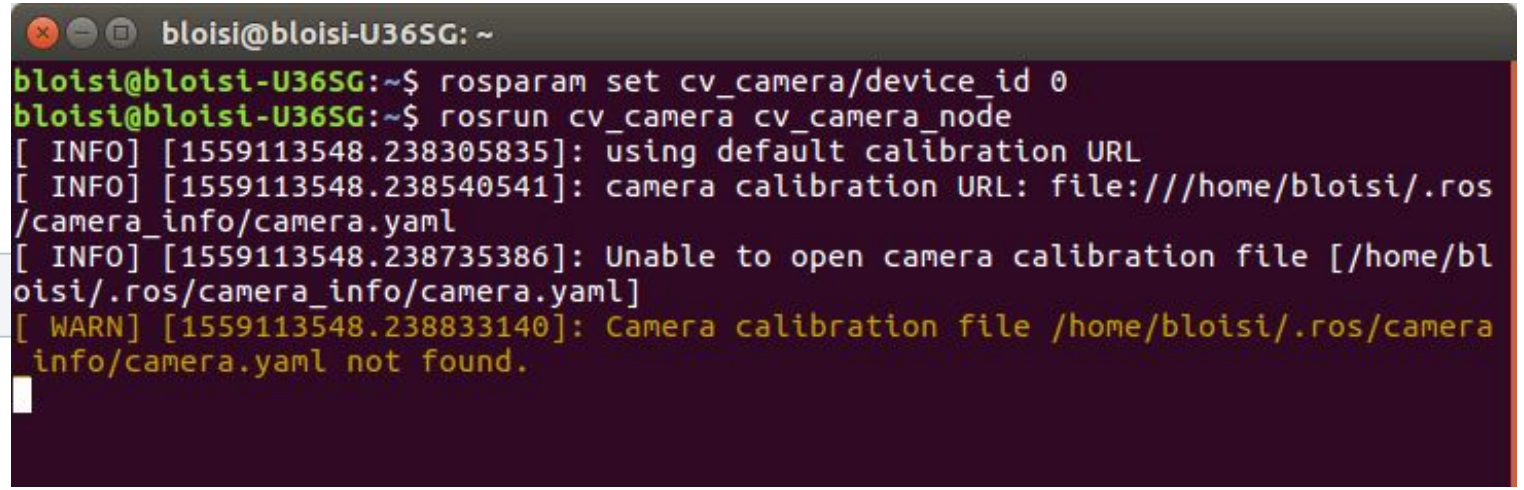
```
rosparam set cv_camera/device_id 0
rosrun cv_camera cv_camera_node
```

- with the first line set the camera device to use
- run the node with the second line

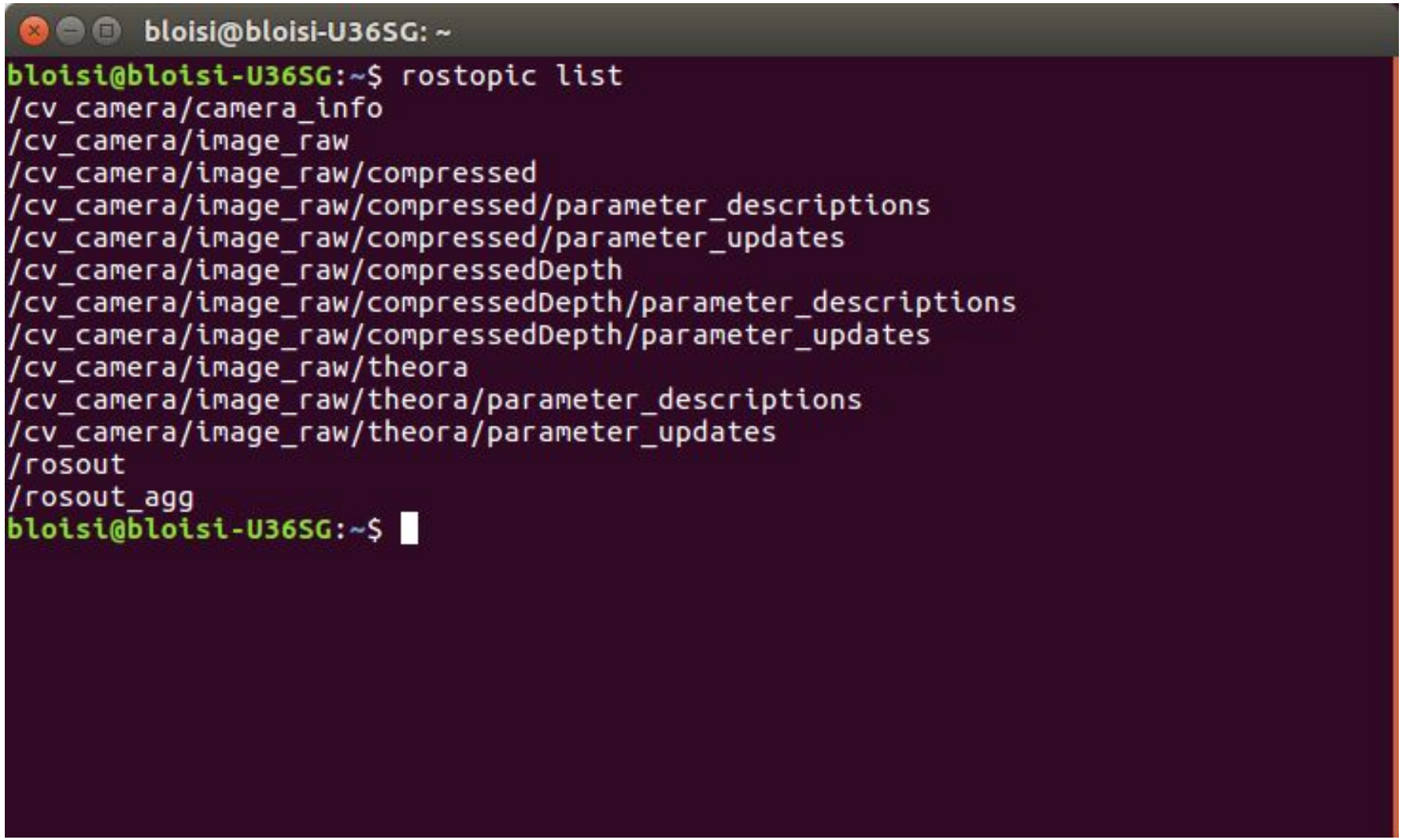
2. Nodelet

This node works as nodelet (cv_camera/CvCameraNodelet).

Wiki: cv_camera (last edited 2017-06-22 11:16:57 by SilvanHeim)



topic di cv_camera



modifichiamo face_detector_node

face_detector_node.py (~/catkin_ws/src/unibas_face_detector/src) - gedit

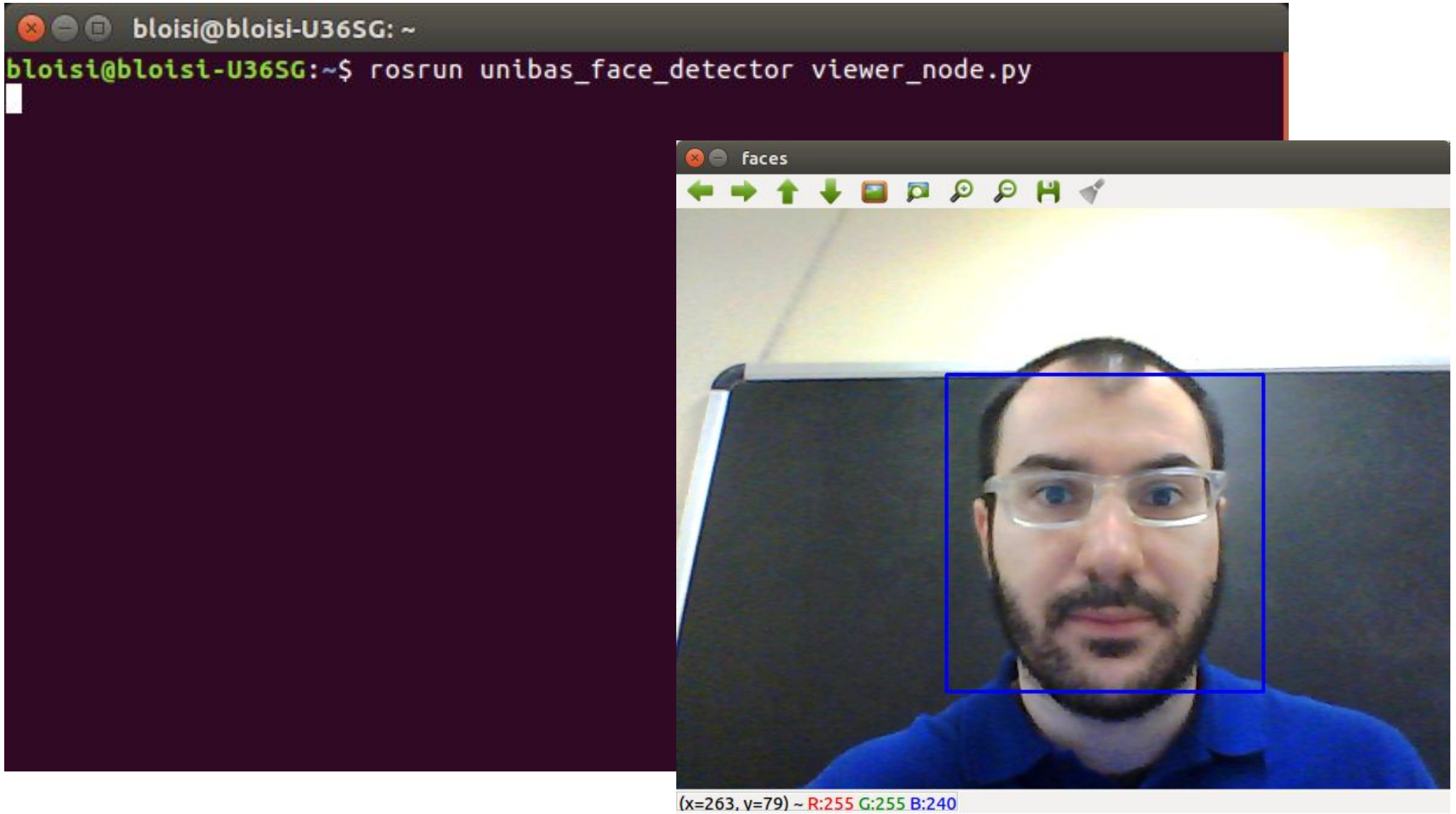
Open Save

```
12 from sensor_msgs.msg import Image
13 from cv_bridge import CvBridge, CvBridgeError
14
15 class face_detector:
16
17   def __init__(self):
18     self.bridge = CvBridge()
19
20     self.image_sub = rospy.Subscriber("/cv_camera/image_raw", Image, self.callback)
21     #self.image_sub = rospy.Subscriber("/camera/rgb/image_raw", Image, self.callback)
22
23     self.pub = rospy.Publisher('/unibas_face_detector/faces', Image, queue_size=1)
24
```

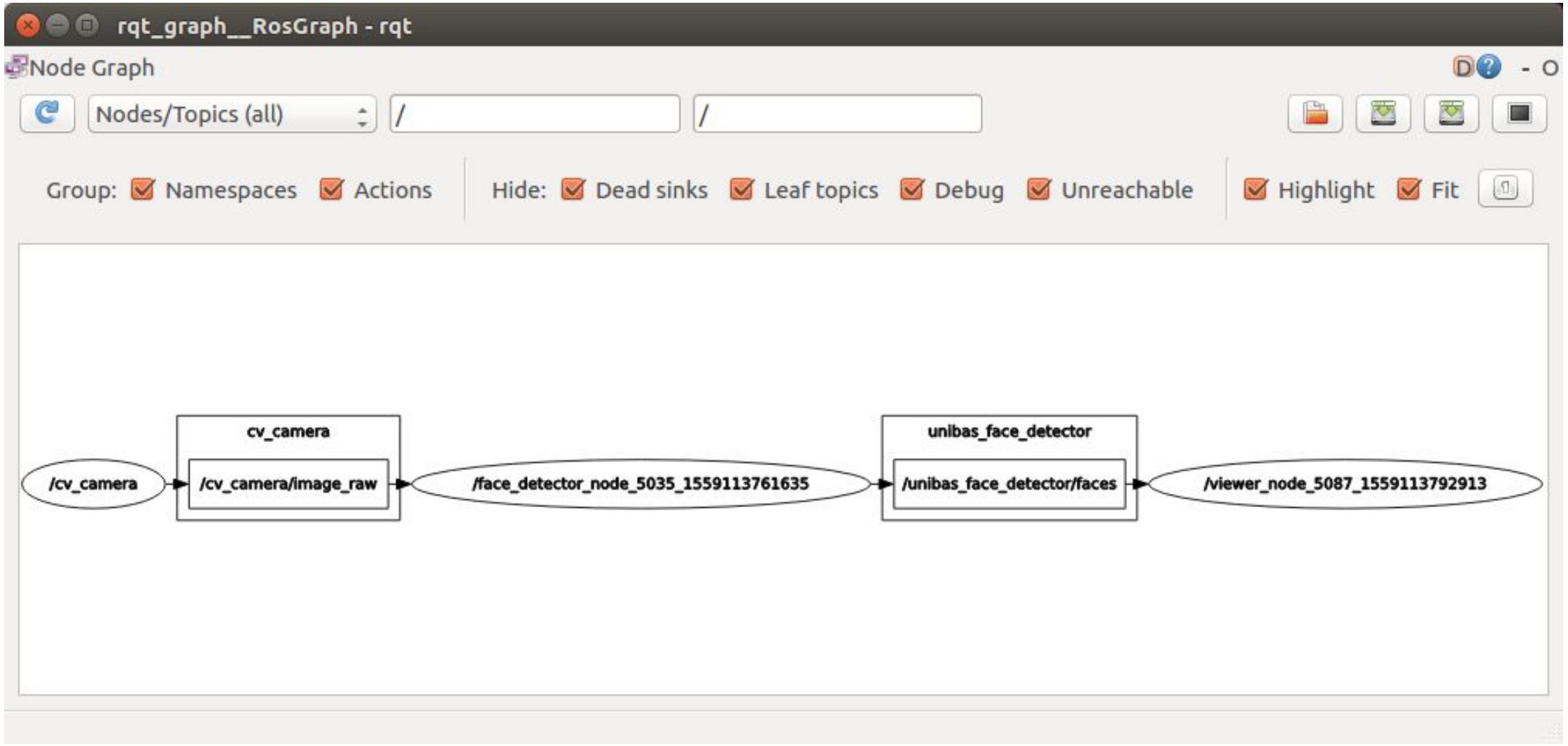
Python Tab Width: 8 Ln 20, Col 51 INS

lanciamo il nodo face_detector_node

lanciamo il nodo viewer_node



rqt_graph


rqt_graph__RosGraph - rqt
Node Graph
Nodes/Topics (all)
/
/
Group: Namespaces Actions
Hide: Dead sinks Leaf topics Debug Unreachable
Highlight Fit
cv_camera
/cv_camera
/cv_camera/image_raw
/face_detector_node_5035_1559113761635
unibas_face_detector
/unibas_face_detector/faces
/viewer_node_5087_1559113792913

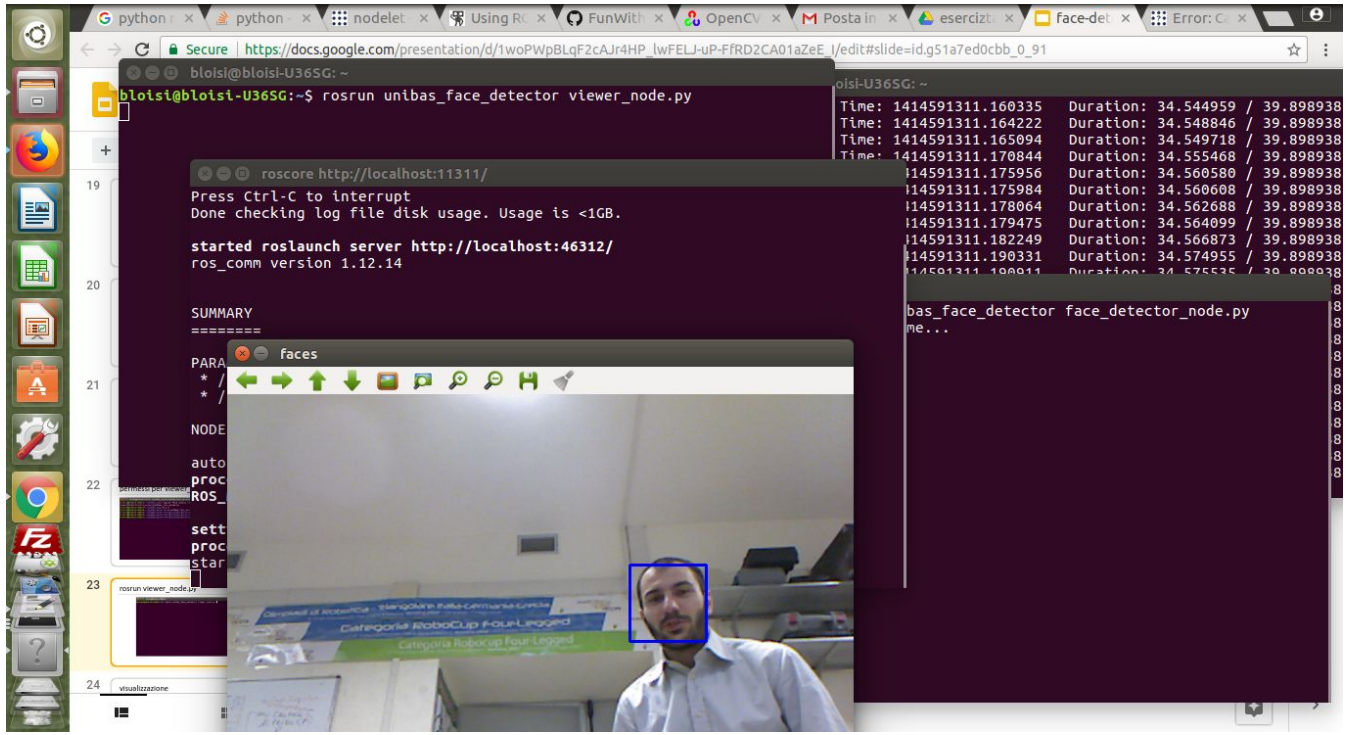
Esercizio 1

Utilizzare la rosbag people.bag
https://drive.google.com/file/d/1oOMahlPdlwJkHMqXLtrLMktfx68-AGfJ/view?usp=sharing
con il package unibas_face_detector

La detection dei volti presenti nella scena è corretta?

Esercizio 2

Creare un launchfile per evitare di dover aprire quattro differenti terminal per utilizzare il package unibas_face_detector

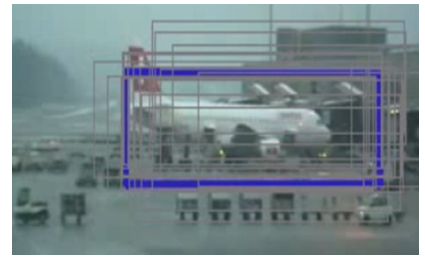


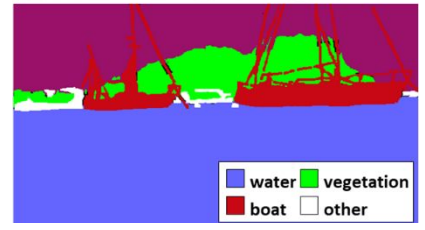
Esercizio 3

Provare ad individuare anche gli occhi e la bocca all'interno della roi del volto come indicato nel tutorial OpenCV al seguente indirizzo
https://docs.opencv.org/3.3.1/d7/d8b/tutorial_py_face_detection.html









UNIVERSITÀ DEGLI STUDI DELLA BASILICATA

Corso di Sistemi Informativi
A.A. 2018/19

Docente
Domenico Daniele Bloisi

face detection

Maggio 2019