

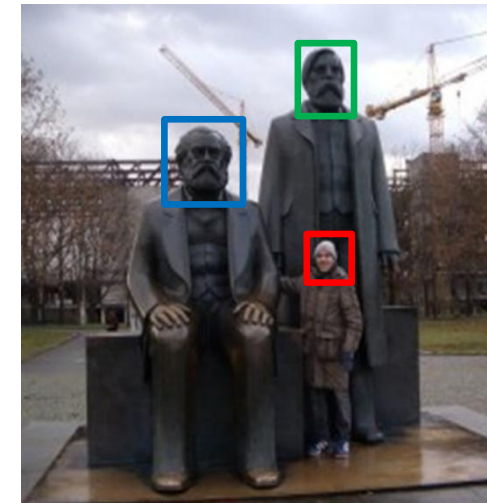
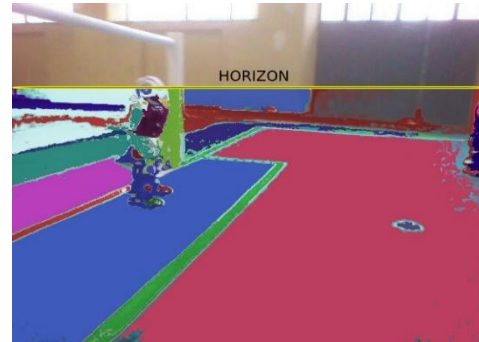
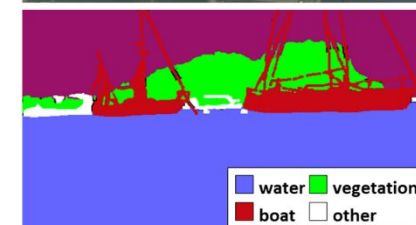


**UNIVERSITÀ DEGLI STUDI
DELLA BASILICATA**

Corso di Sistemi Informativi
A.A. 2018/19

Docente
Domenico Daniele Bloisi

face detection



Giugno 2019

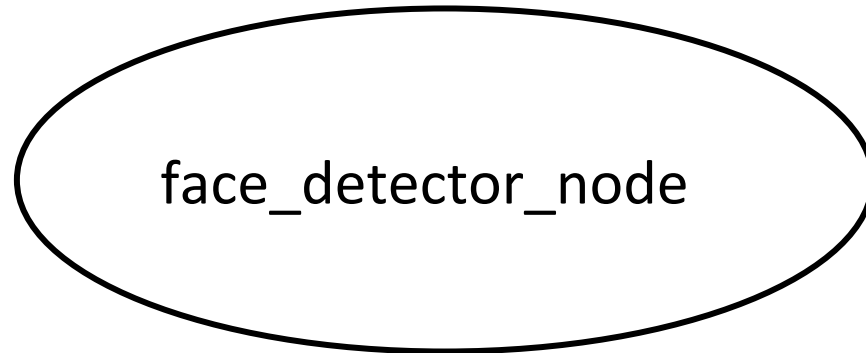
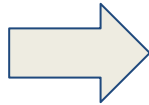
obiettivo

- 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

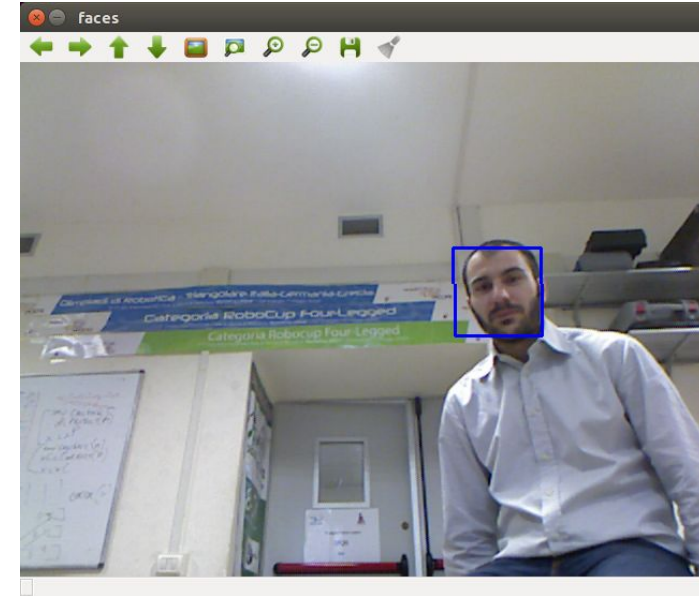
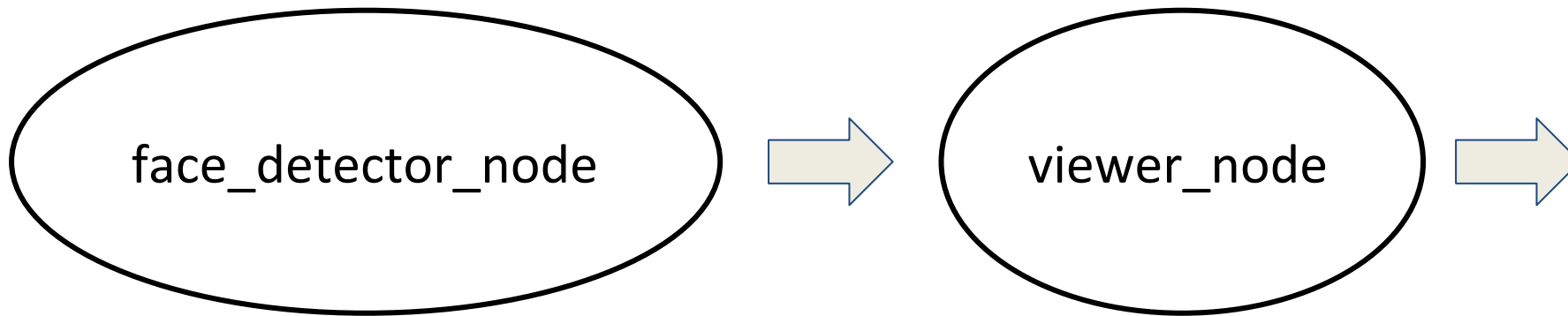
Package unibas_face_detector



rosvag acquisita con
una telecamera



face_detector_node e viewer_node

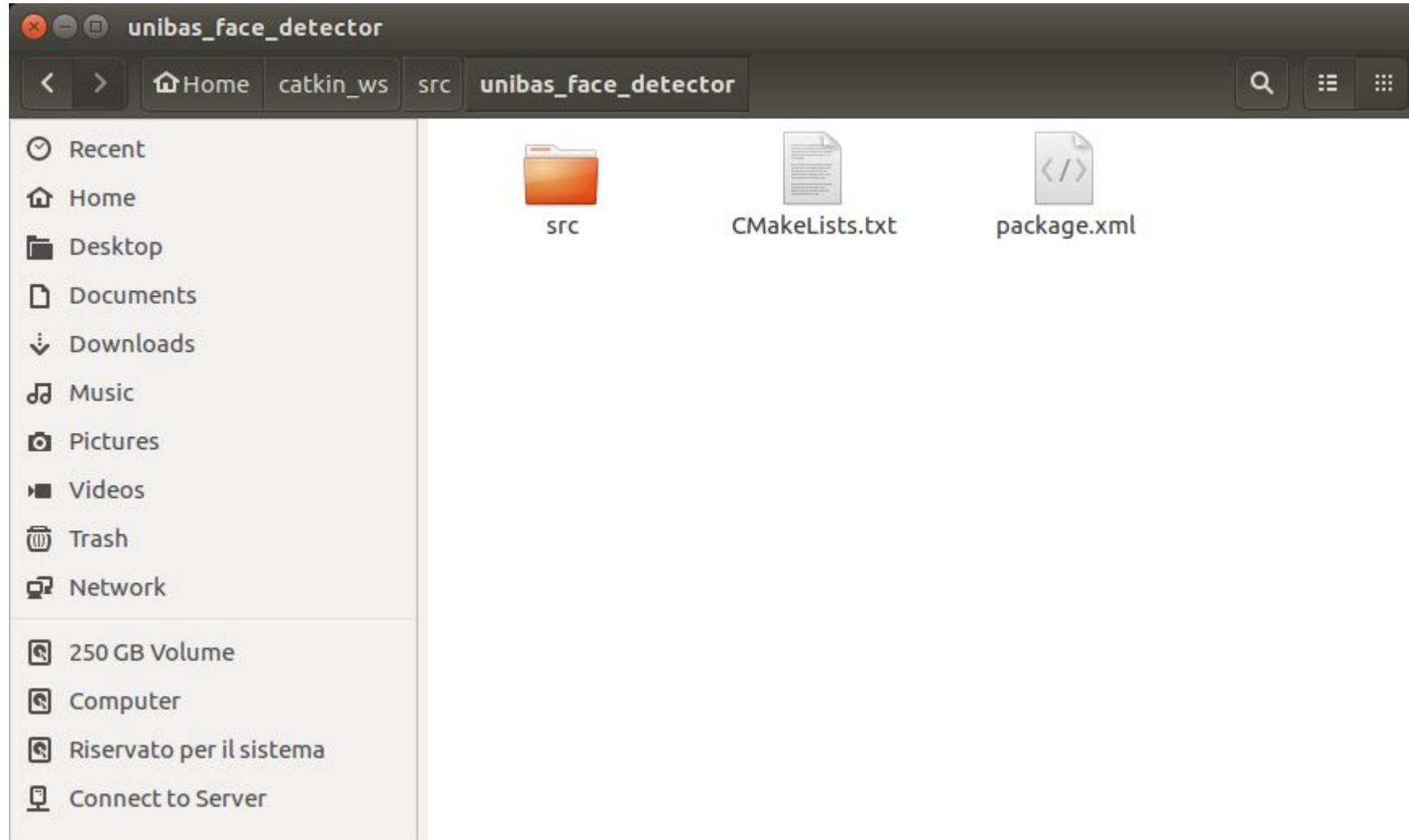


visualizzazione
immagine OpenCV

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$
```


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. Please 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=/home/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
```

```
bloisi@bloisi-U36SG: ~/catkin_ws
[ 52%] Built target hw1_generate_messages_py
[ 53%] Built target hw1_generate_messages_lisp
[ 53%] Built target turtlebot3_msgs_generate_messages_eus
[ 61%] Built target turtlebot3_applications_msgs_generate_messages_py
[ 63%] Built target turtlebot3_applications_msgs_generate_messages_cpp
[ 65%] Built target turtlebot3_applications_msgs_generate_messages_lisp
[ 70%] Built target turtlebot3_example_generate_messages_py
[ 75%] Built target turtlebot3_example_generate_messages_nodejs
[ 79%] Built target turtlebot3_example_generate_messages_cpp
[ 81%] Built target turtlebot3_applications_msgs_generate_messages_nodejs
[ 87%] Built target turtlebot3_example_generate_messages_eus
[ 89%] Built target turtlebot3_diagnostics
[ 94%] Built target turtlebot3_example_generate_messages_lisp
[ 94%] Built target turtlebot3_msgs_generate_messages
[ 96%] Built target turtlebot3_fake_node
[ 97%] Built target homework_1_generate_messages
[ 97%] Built target turtlebot3_drive
[100%] Built target turtlebot3_panorama
[100%] Built target hw1_generate_messages
[100%] Built target turtlebot3_example_generate_messages
[100%] Built target turtlebot3_applications_msgs_generate_messages
bloisi@bloisi-U36SG:~/catkin_ws$ . ~/catkin_ws/devel/setup.bash
bloisi@bloisi-U36SG:~/catkin_ws$
```


rospack find

```
bloisi@bloisi-U36SG: ~/catkin_ws
bloisi@bloisi-U36SG:~/catkin_ws$ rospack find unibas_face_detector
/home/bloisi/catkin_ws/src/unibas_face_detector
bloisi@bloisi-U36SG:~/catkin_ws$
```

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 ▾ [icon] 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
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

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



```
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

```
bloisi@bloisi-U36SG: ~/catkin_ws/src/unibas_face_detector/src
bloisi@bloisi-U36SG:~/catkin_ws$ rospack find unibas_face_detector
/home/bloisi/catkin_ws/src/unibas_face_detector
bloisi@bloisi-U36SG:~/catkin_ws$ cd src
bloisi@bloisi-U36SG:~/catkin_ws/src$ cd unibas_face_detector/
bloisi@bloisi-U36SG:~/catkin_ws/src/unibas_face_detector$ cd src/
bloisi@bloisi-U36SG:~/catkin_ws/src/unibas_face_detector/src$ chmod +x face_detector_node.py
```


roscore

```
roscore http://localhost:11311/
bloisi@bloisi-U36SG:~$ roscore
... logging to /home/bloisi/.ros/log/78cf387c-7bbf-11e9-b0ad-50465dde6884/roslau
nch-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
=====

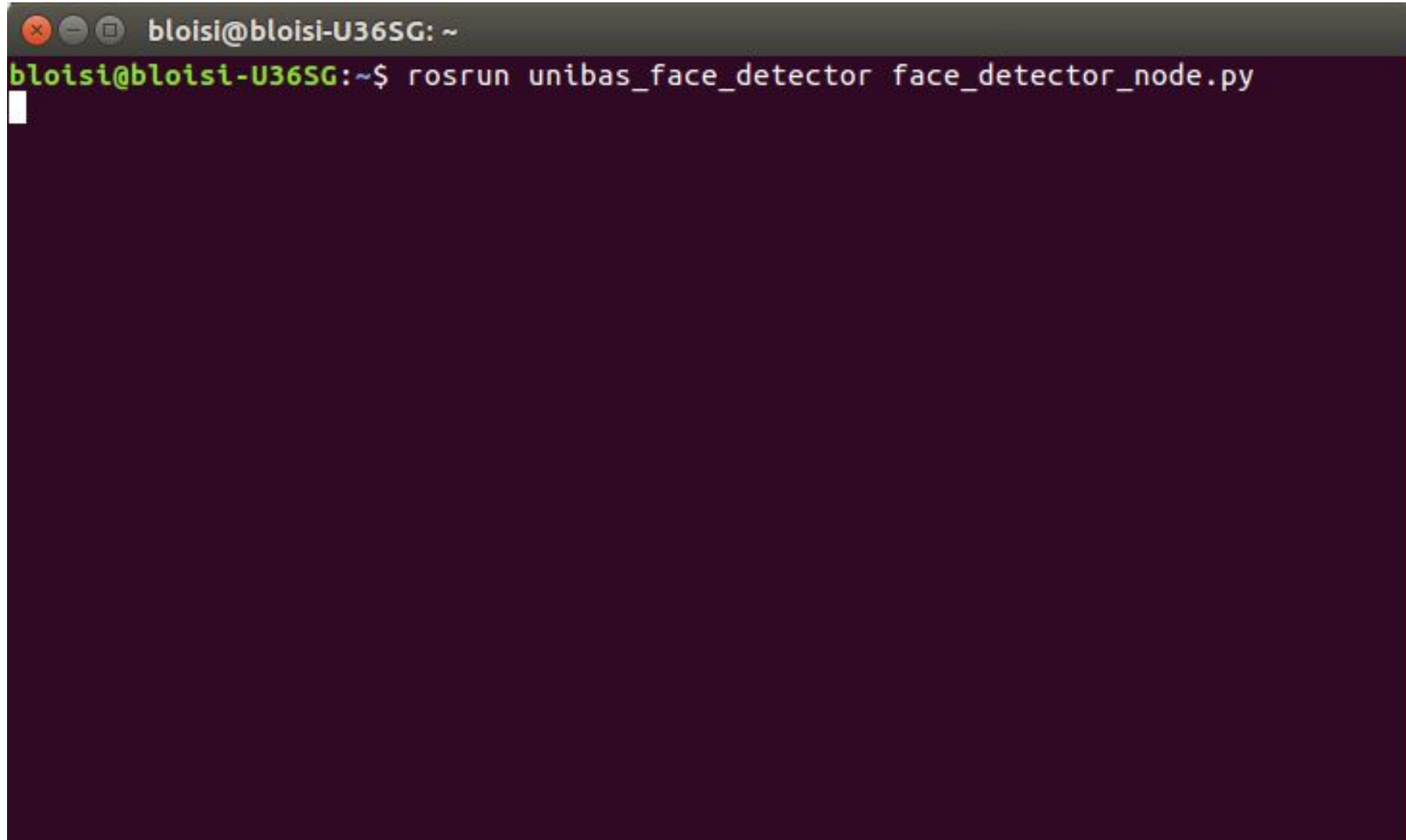
PARAMETERS
* /rostdistro: kinetic
* /rosversion: 1.12.14

NODES

auto-starting new master
process[master]: started with pid [8584]
ROS_MASTER_URI=http://localhost:11311/

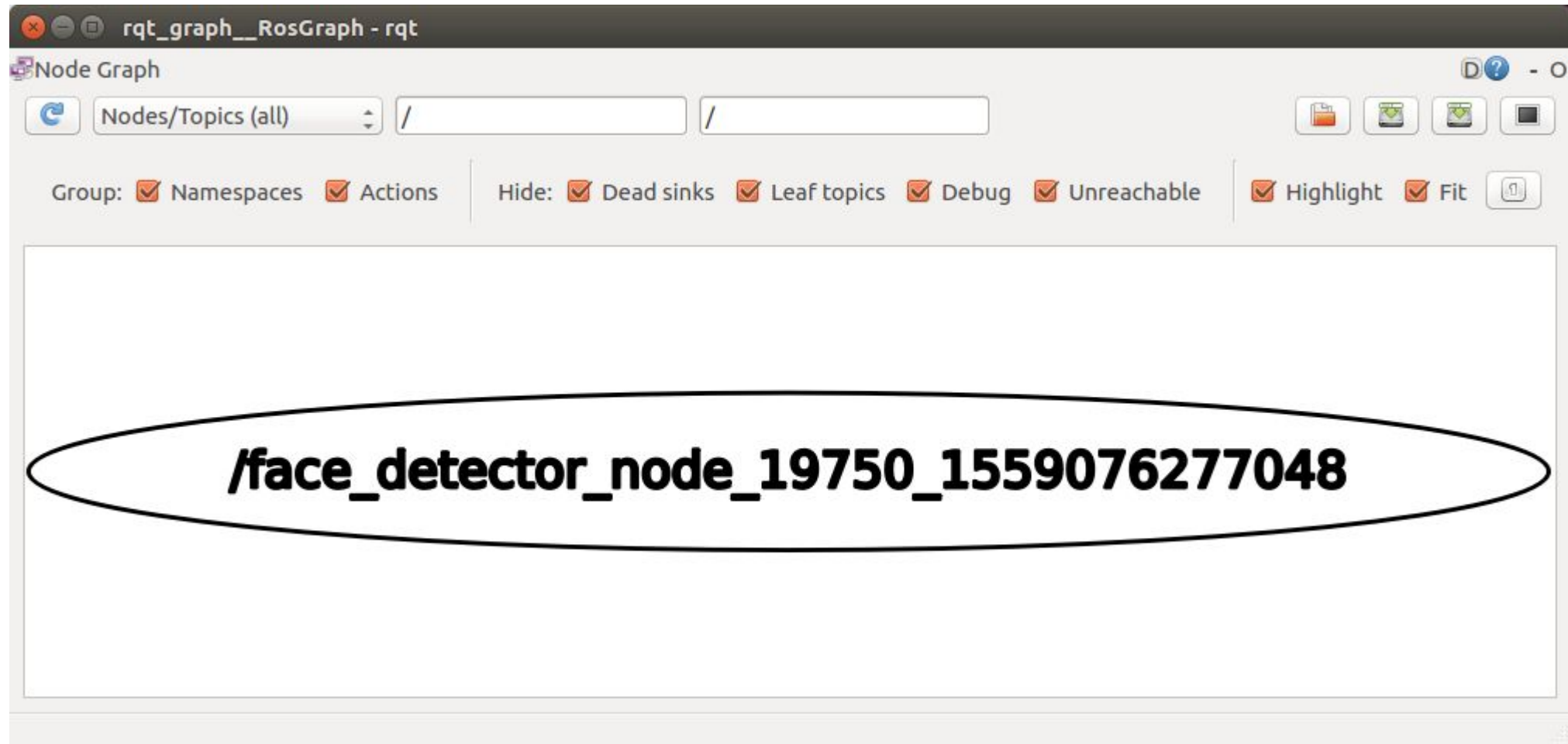
setting /run_id to 78cf387c-7bbf-11e9-b0ad-50465dde6884
process[rosout-1]: started with pid [8733]
started core service [/rosout]
```

roslaunch face_detector_node

A terminal window with a dark purple background and a grey title bar. The title bar contains three window control icons (close, minimize, maximize) and the text 'bloisi@bloisi-U36SG: ~'. The terminal shows a command prompt 'bloisi@bloisi-U36SG:~\$' followed by the command 'roslaunch unibas_face_detector face_detector_node.py'. A white cursor is positioned at the end of the command.

```
bloisi@bloisi-U36SG: ~  
bloisi@bloisi-U36SG:~$ roslaunch unibas_face_detector face_detector_node.py
```

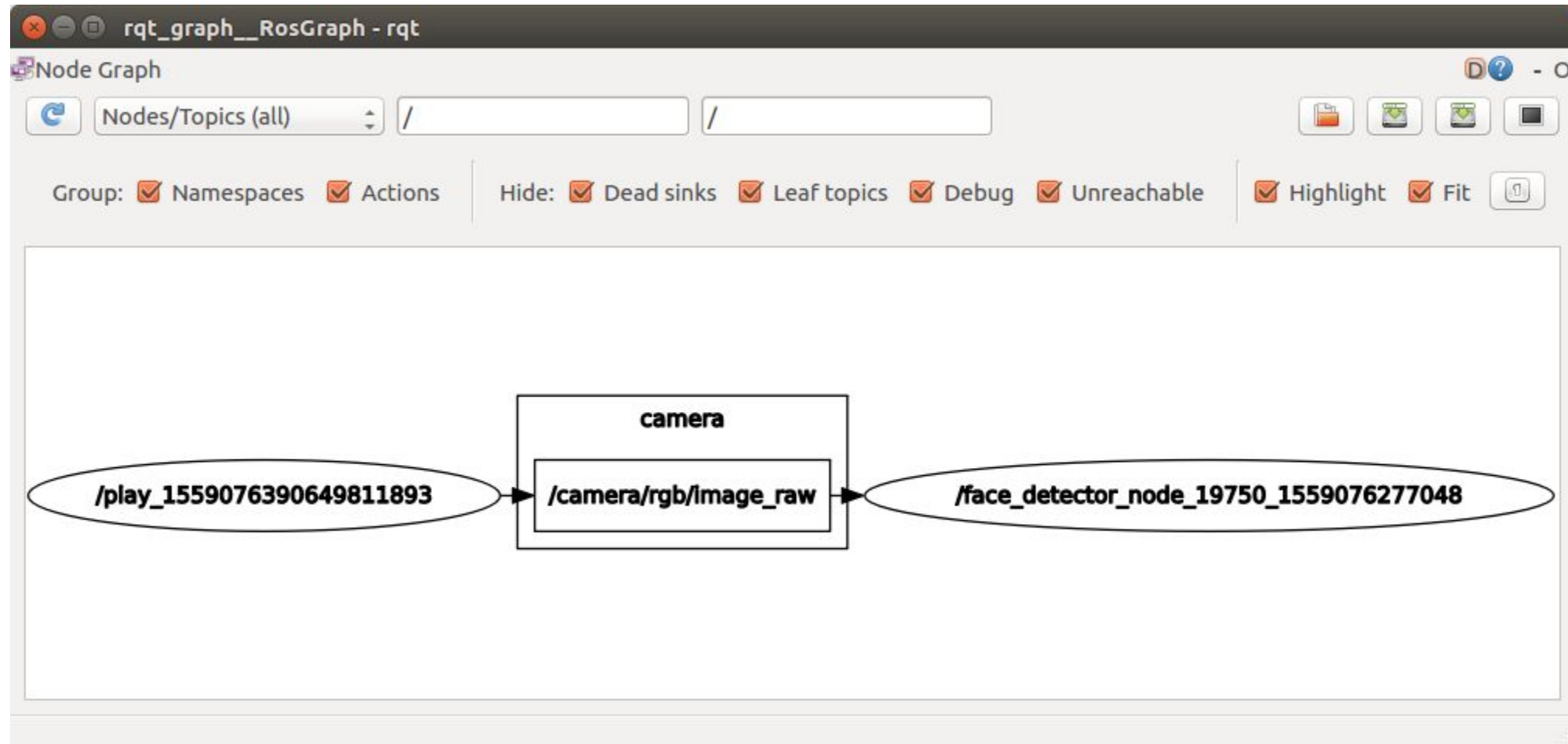
rqt_graph



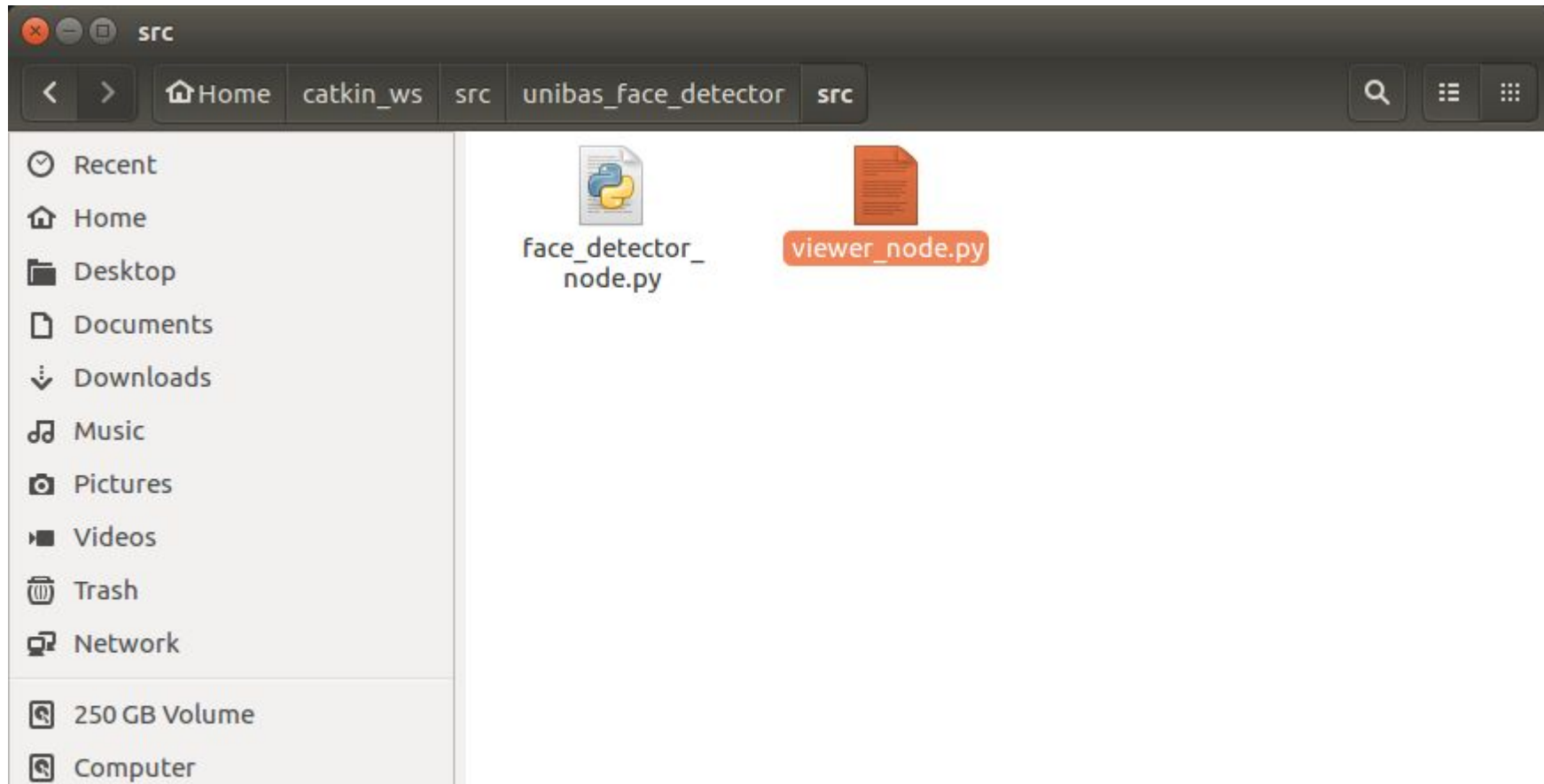
rosvag play

```
bloisi@bloisi-U36SG: ~  
bloisi@bloisi-U36SG:~$ rosvag 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
```

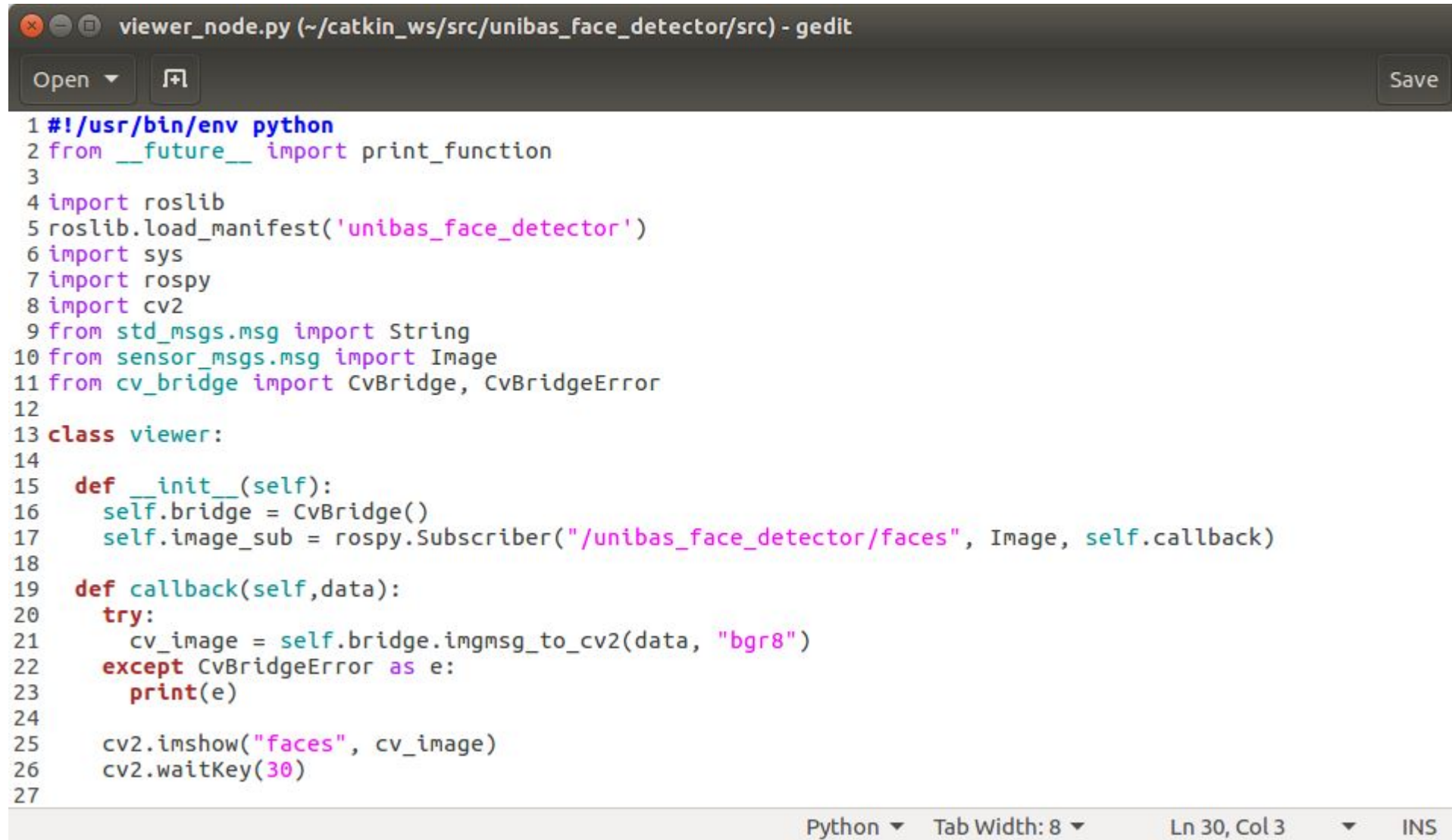
rqt_graph



viewer_node



codice viewer_node



```
1#!/usr/bin/env python
2from __future__ import print_function
3
4import roslib
5roslib.load_manifest('unibas_face_detector')
6import sys
7import rospy
8import cv2
9from std_msgs.msg import String
10from sensor_msgs.msg import Image
11from cv_bridge import CvBridge, CvBridgeError
12
13class 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

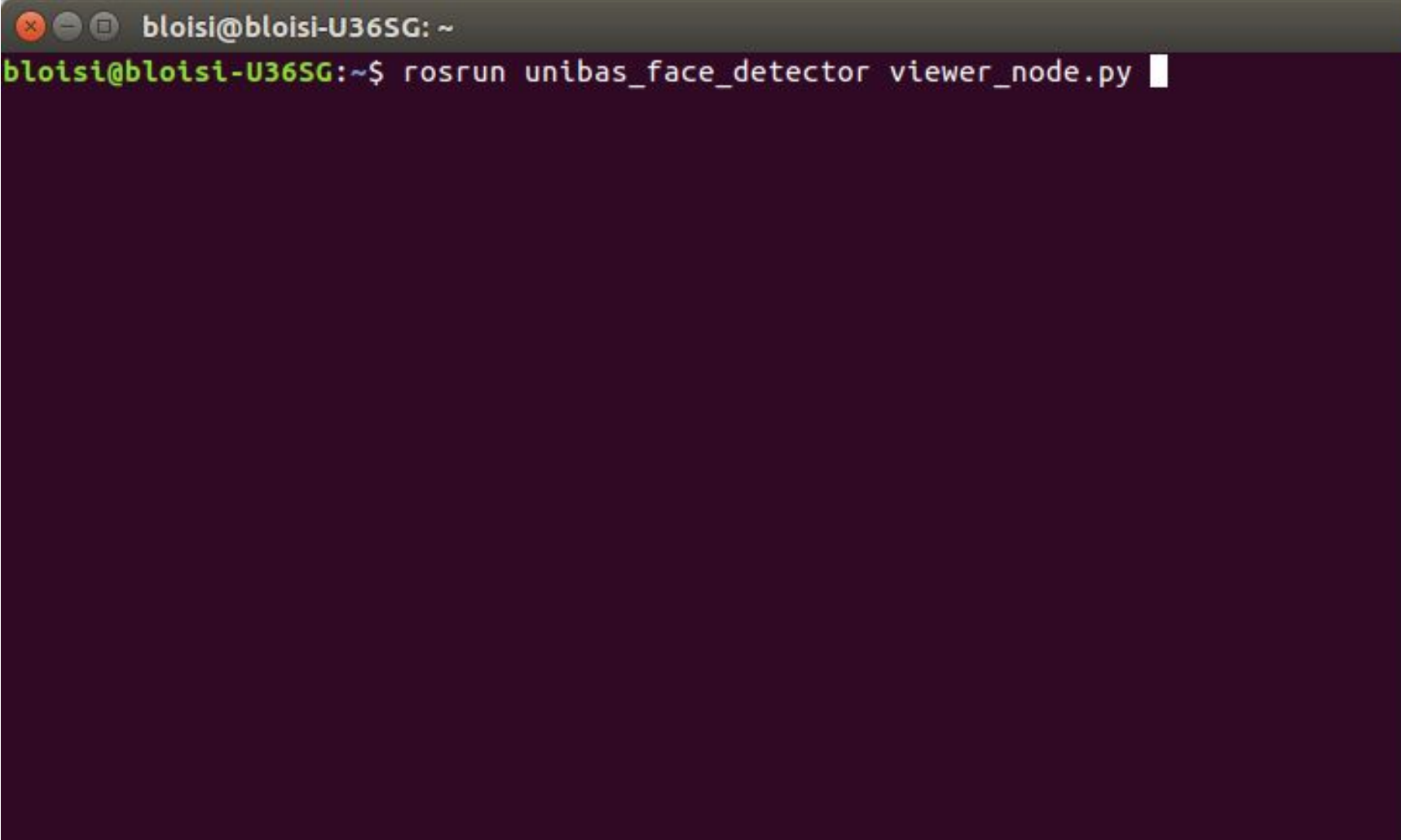


```
viewer_node.py (~/.catkin_ws/src/unibas_face_detector/src) - gedit
Open Save
28
29 def main(args):
30     v = viewer()
31     rospy.init_node('viewer_node', anonymous=True)
32     try:
33         rospy.spin()
34     except KeyboardInterrupt:
35         print("Shutting down")
36     cv2.destroyAllWindows()
37
38 if __name__ == '__main__':
39     main(sys.argv)
40
Python Tab Width: 8 Ln 30, Col 3 INS
```

permessi per viewer_node.py

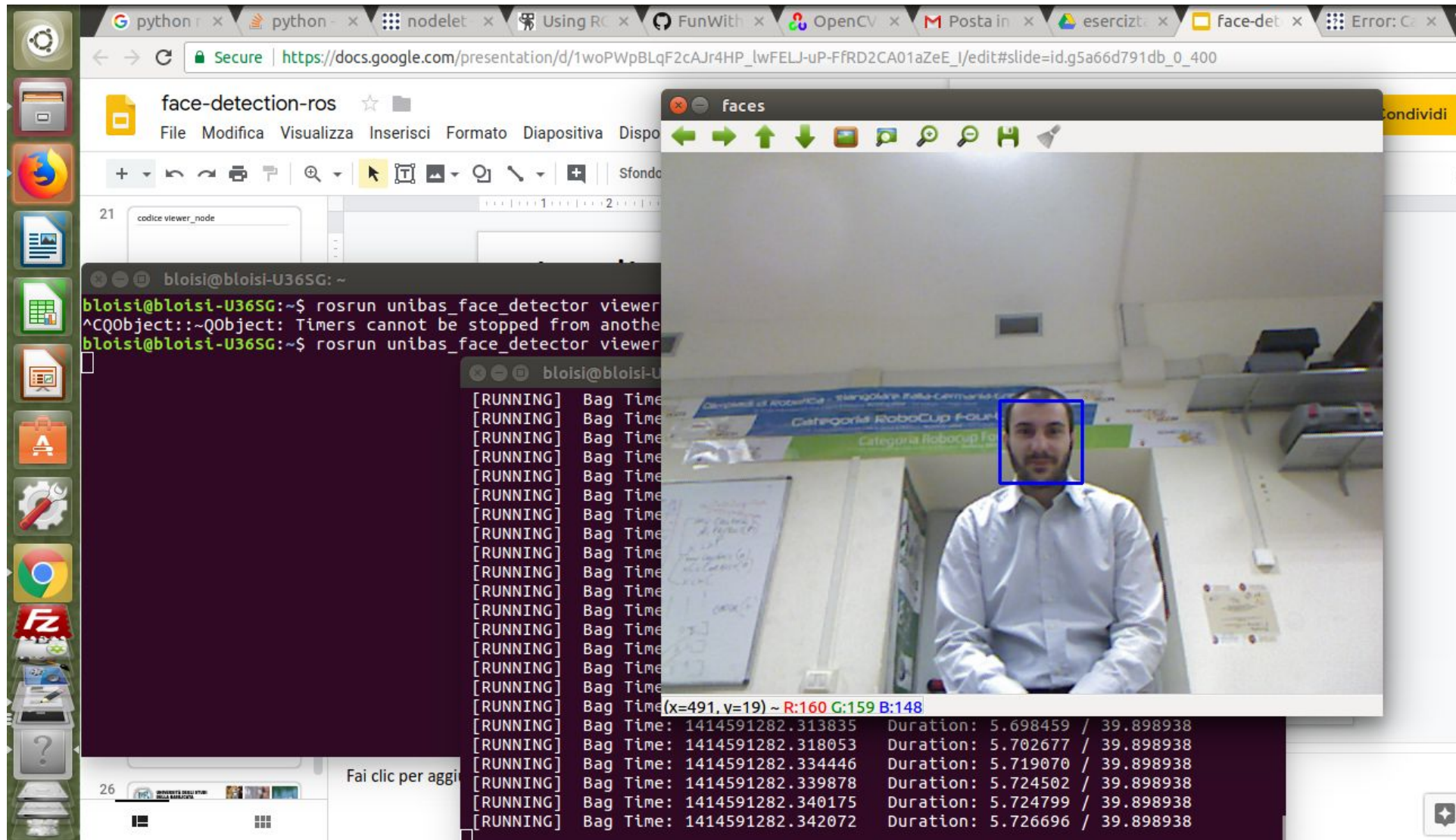
```
bloisi@bloisi-U36SG: ~/catkin_ws/src/unibas_face_detector/src
bloisi@bloisi-U36SG:~/catkin_ws$ rospack find unibas_face_detector
/home/bloisi/catkin_ws/src/unibas_face_detector
bloisi@bloisi-U36SG:~/catkin_ws$ cd src
bloisi@bloisi-U36SG:~/catkin_ws/src$ cd unibas_face_detector/
bloisi@bloisi-U36SG:~/catkin_ws/src/unibas_face_detector/src$ chmod +x face_detector_node.py
bloisi@bloisi-U36SG:~/catkin_ws/src/unibas_face_detector/src$ chmod +x viewer_node.py
bloisi@bloisi-U36SG:~/catkin_ws/src/unibas_face_detector/src$
```

roslaunch viewer_node.py

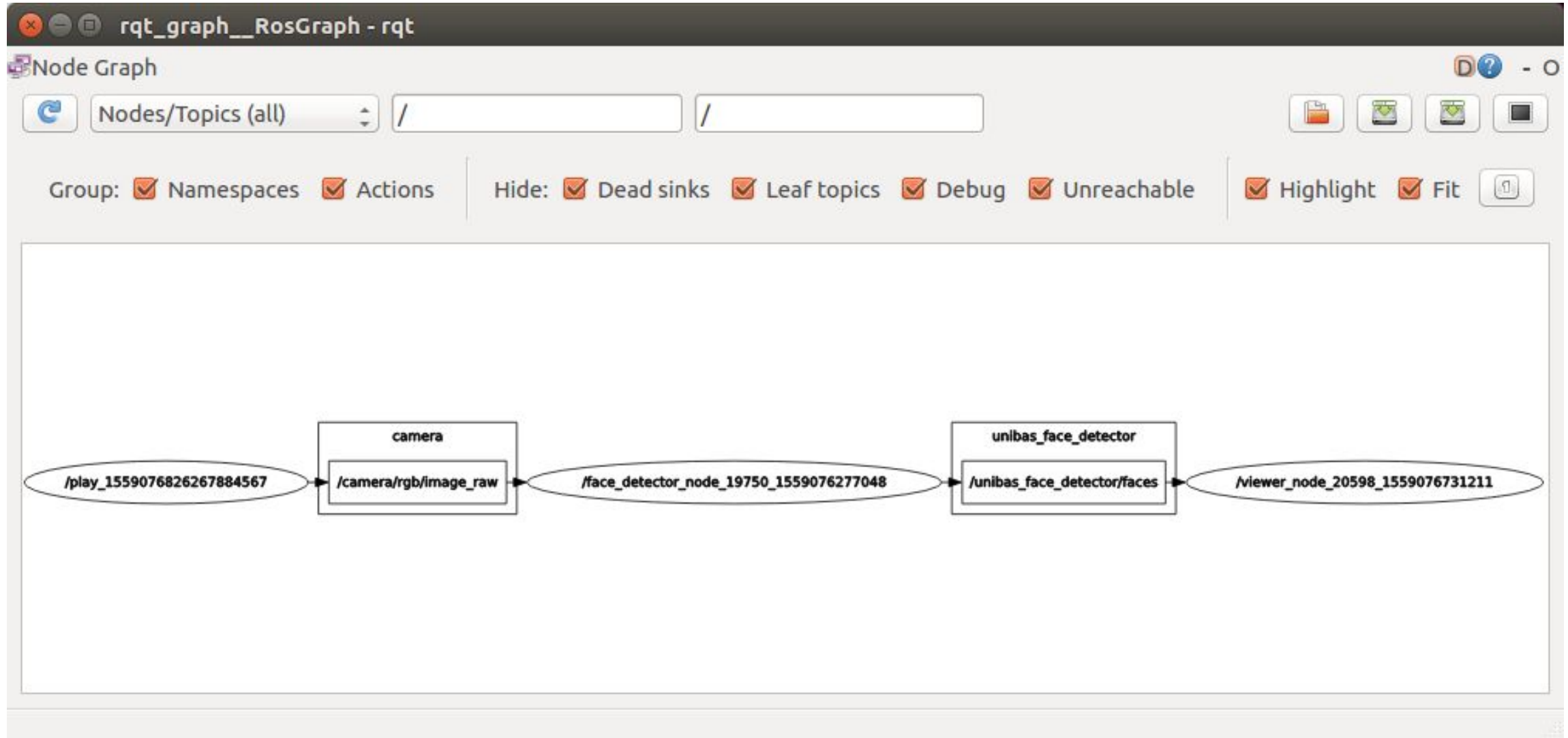
A terminal window with a dark purple background and a grey title bar. The title bar contains the text 'bloisi@bloisi-U36SG: ~' and standard window control buttons. The terminal shows a green prompt 'bloisi@bloisi-U36SG:~\$' followed by the command 'roslaunch unibas_face_detector viewer_node.py' in white text. A white cursor is at the end of the command.

```
bloisi@bloisi-U36SG: ~  
bloisi@bloisi-U36SG:~$ roslaunch unibas_face_detector viewer_node.py
```


visualizzazione



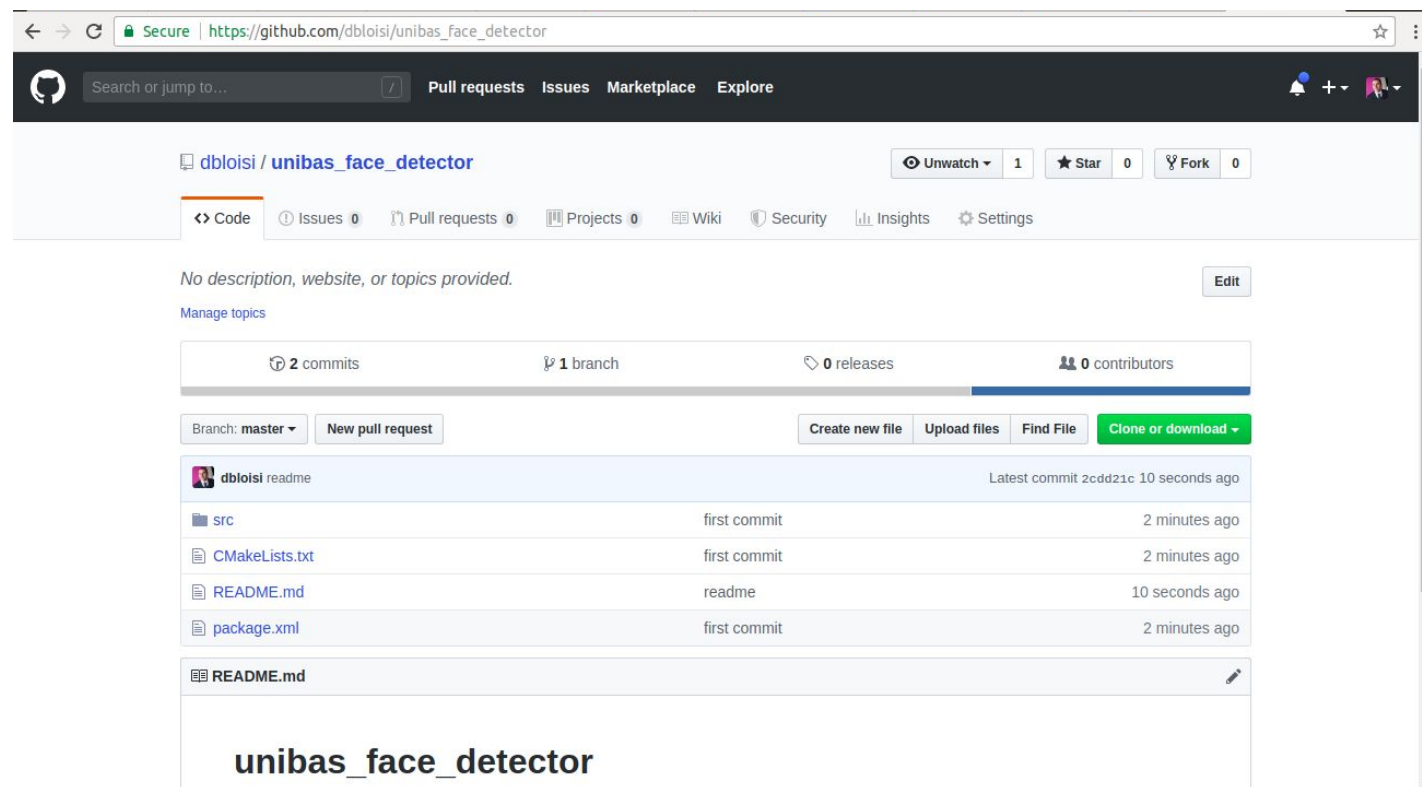
rqt_graph



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

[←](#) [→](#) [↻](#) [wiki.ros.org/cv_camera](#) [☆](#) [⋮](#)

ROSCon 2019 Diversity Scholarships: Applications Open!
Whoever you are, whatever you do, and wherever you do it, if you're interested in ROS, then we want you to join us at ROSCon in Macau! [×](#)

ROS.org [About](#) | [Support](#) | [Discussion Forum](#) | [Service Status](#) | [Q&A answers.ros.org](#) Search: [Submit](#)

Documentation **Browse Software** **News** **Download**

cv_camera
[indigo](#) [kinetic](#) [lunar](#) [melodic](#) [Show EOL distros:](#) ☐

[Documentation Status](#)

Package Summary

✓ Released ✓ Continuous Integration ✓ Documented

cv_camera uses OpenCV capture object to capture camera image. This supports camera_image and nodelet.

- Maintainer status: developed
- Maintainer: Takashi Ogura <t.ogura AT gmail DOT com>
- Author: Takashi Ogura <t.ogura AT gmail DOT com>
- License: BSD
- Source: git https://github.com/OTL/cv_camera.git (branch: master)

Contents

- 1. [ROS OpenCV camera driver](#)
 - 1. [cv_camera_node](#)
 - 1. [Publish](#)
 - 2. [Service](#)

Package Links

- [Code API](#)
- [FAQ](#)
- [Changelog](#)
- [Change List](#)
- [Reviews](#)
- [Dependencies \(9\)](#)
- [Jenkins jobs \(7\)](#)

Wiki

- [Distributions](#)
- [ROS/Installation](#)
- [ROS/Tutorials](#)
- [RecentChanges](#)
- [cv_camera](#)

Page

- [Immutable Page](#)
- [Info](#)
- [Attachments](#)
- [More Actions:](#) ▼

User

- [Login](#)

installazione package cv_camera

```
bloisi@bloisi-U36SG: ~  
bloisi@bloisi-U36SG:~$ sudo apt-get install ros-kinetic-cv-camera  
[sudo] password for bloisi:  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
The following packages were automatically installed and are no longer required:  
  linux-headers-4.4.0-128-generic linux-image-4.4.0-124-generic  
  linux-image-4.4.0-127-generic linux-image-4.4.0-128-generic  
  linux-image-extra-4.4.0-124-generic linux-image-extra-4.4.0-127-generic  
  linux-image-extra-4.4.0-128-generic snapd-login-service  
Use 'sudo apt autoremove' to remove them.  
The following NEW packages will be installed:  
  ros-kinetic-cv-camera  
0 upgraded, 1 newly installed, 0 to remove and 62 not upgraded.  
Need to get 50,6 kB of archives.  
After this operation, 219 kB of additional disk space will be used.  
Get:1 http://packages.ros.org/ros/ubuntu xenial/main amd64 ros-kinetic-cv-camera  
  amd64 0.3.0-0xenial-20190320-144216-0800 [50,6 kB]  
Fetched 50,6 kB in 0s (54,2 kB/s)  
Selecting previously unselected package ros-kinetic-cv-camera.  
(Reading database ... 455115 files and directories currently installed.)  
Preparing to unpack .../ros-kinetic-cv-camera_0.3.0-0xenial-20190320-144216-0800
```

`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

```
roscpp set cv_camera/device_id 0  
roscpp 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).

Except where otherwise noted, the ROS wiki is licensed under the [Creative Commons Attribution 3.0](#) | [Find us on Google+](#)

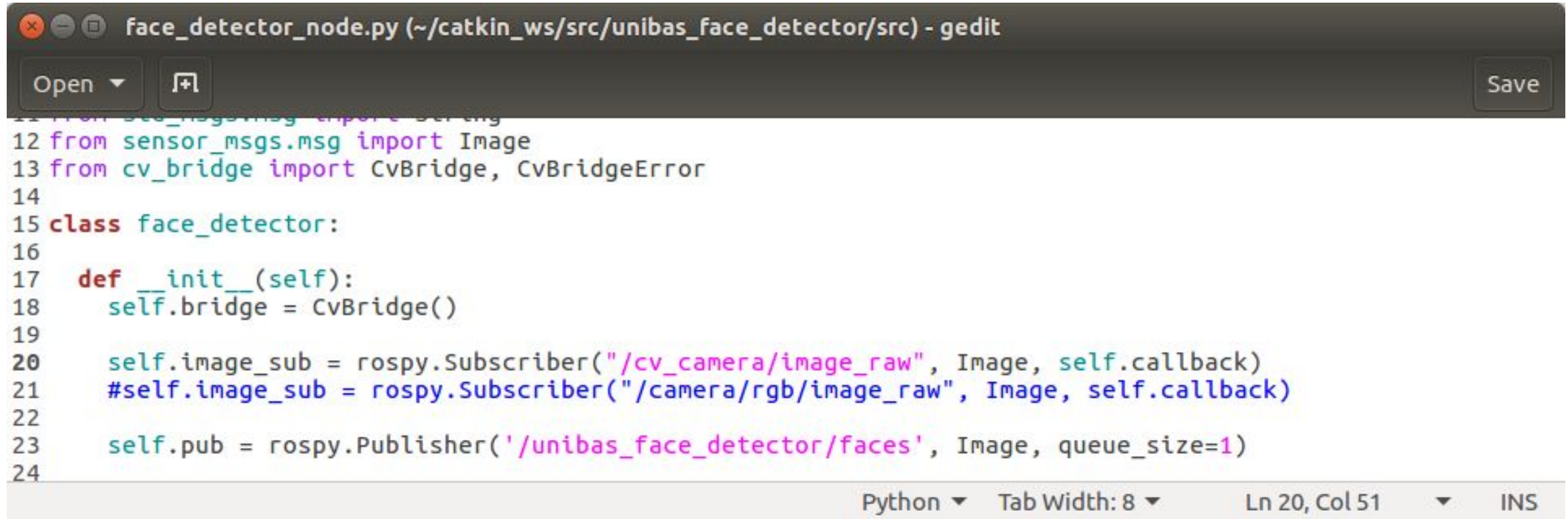
```
bloisi@bloisi-U36SG: ~  
bloisi@bloisi-U36SG:~$ roscpp set cv_camera/device_id 0  
bloisi@bloisi-U36SG:~$ roscpp cv_camera cv_camera_node  
[ INFO] [1559113548.238305835]: using default calibration URL  
[ INFO] [1559113548.238540541]: camera calibration URL: file:///home/bloisi/.ros  
/camera_info/camera.yaml  
[ INFO] [1559113548.238735386]: Unable to open camera calibration file [/home/bl  
oisi/.ros/camera_info/camera.yaml]  
[ WARN] [1559113548.238833140]: Camera calibration file /home/bloisi/.ros/camera  
info/camera.yaml not found.
```

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

topic di cv_camera

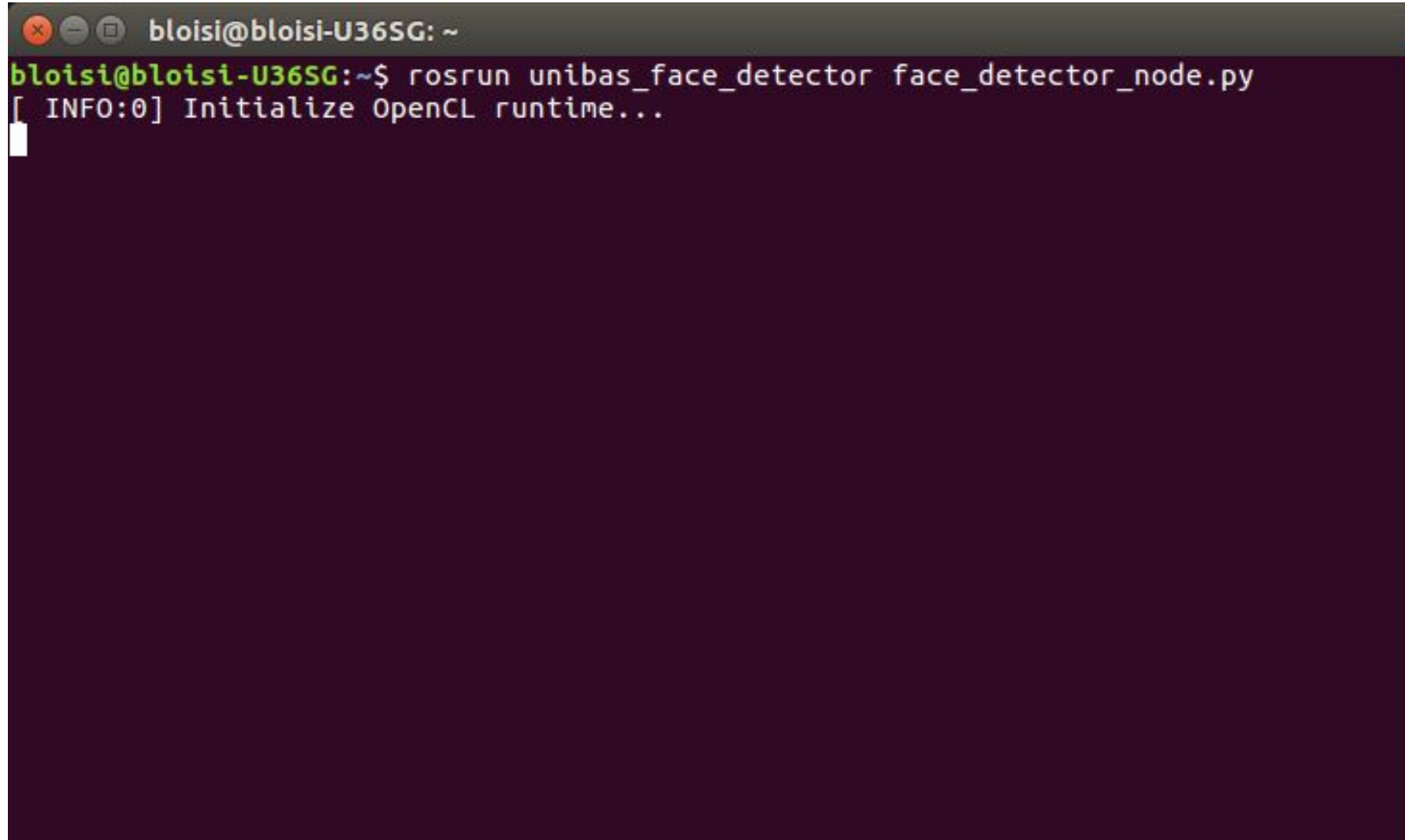
```
bloisi@bloisi-U36SG: ~  
bloisi@bloisi-U36SG:~$ rostopic list  
/cv_camera/camera_info  
/cv_camera/image_raw  
/cv_camera/image_raw/compressed  
/cv_camera/image_raw/compressed/parameter_descriptions  
/cv_camera/image_raw/compressed/parameter_updates  
/cv_camera/image_raw/compressedDepth  
/cv_camera/image_raw/compressedDepth/parameter_descriptions  
/cv_camera/image_raw/compressedDepth/parameter_updates  
/cv_camera/image_raw/theora  
/cv_camera/image_raw/theora/parameter_descriptions  
/cv_camera/image_raw/theora/parameter_updates  
/rosout  
/rosout_agg  
bloisi@bloisi-U36SG:~$
```


modifichiamo face_detector_node



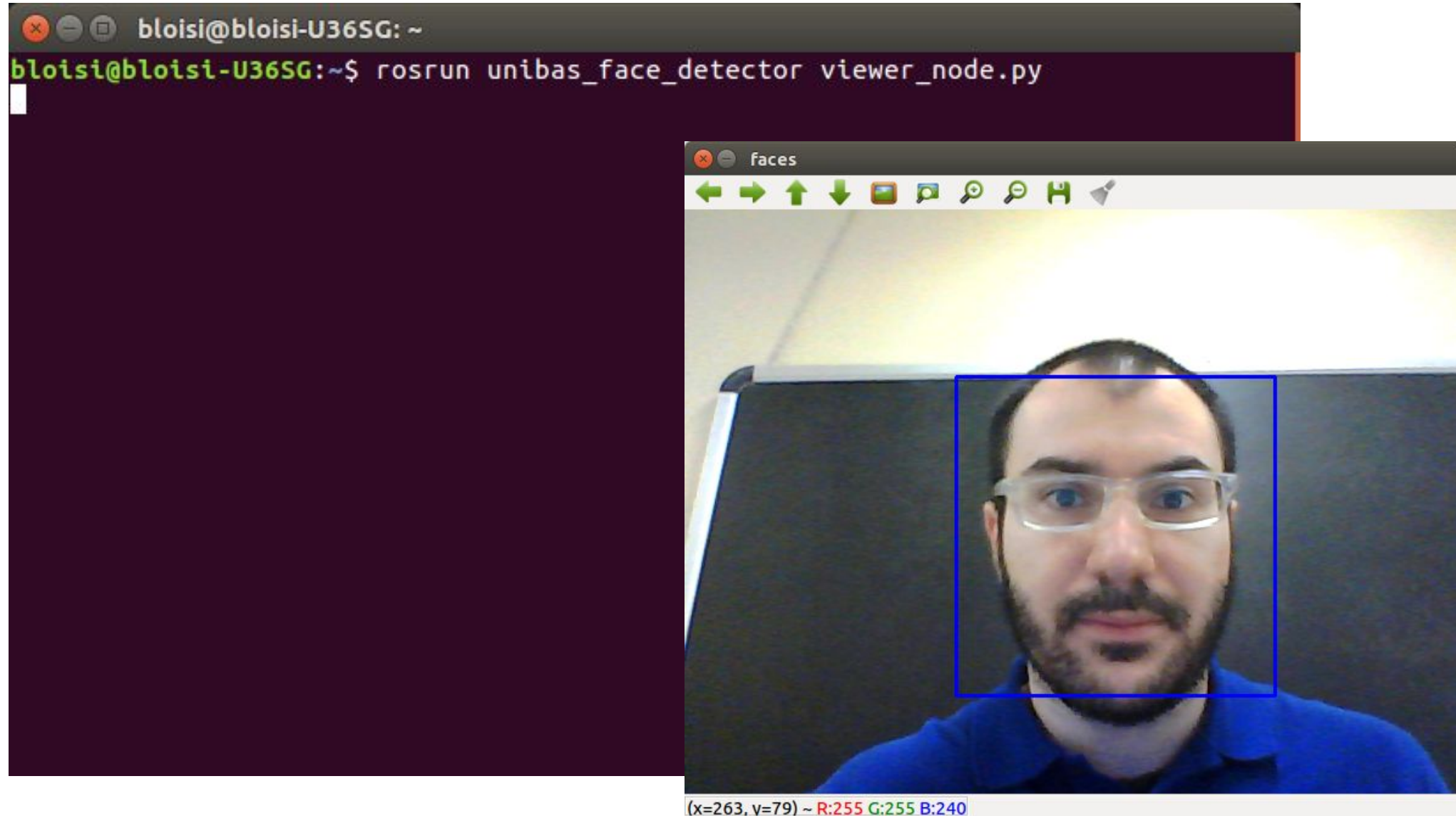
```
face_detector_node.py (~/.catkin_ws/src/unibas_face_detector/src) - gedit
Open Save
11 from sensor_msgs.msg import Image
12 from sensor_msgs.msg import Image
13 from cv_bridge import CvBridge, CvBridgeError
14
15 class face_detector:
16
17     def __init__(self):
18         self.bridge = CvBridge()
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

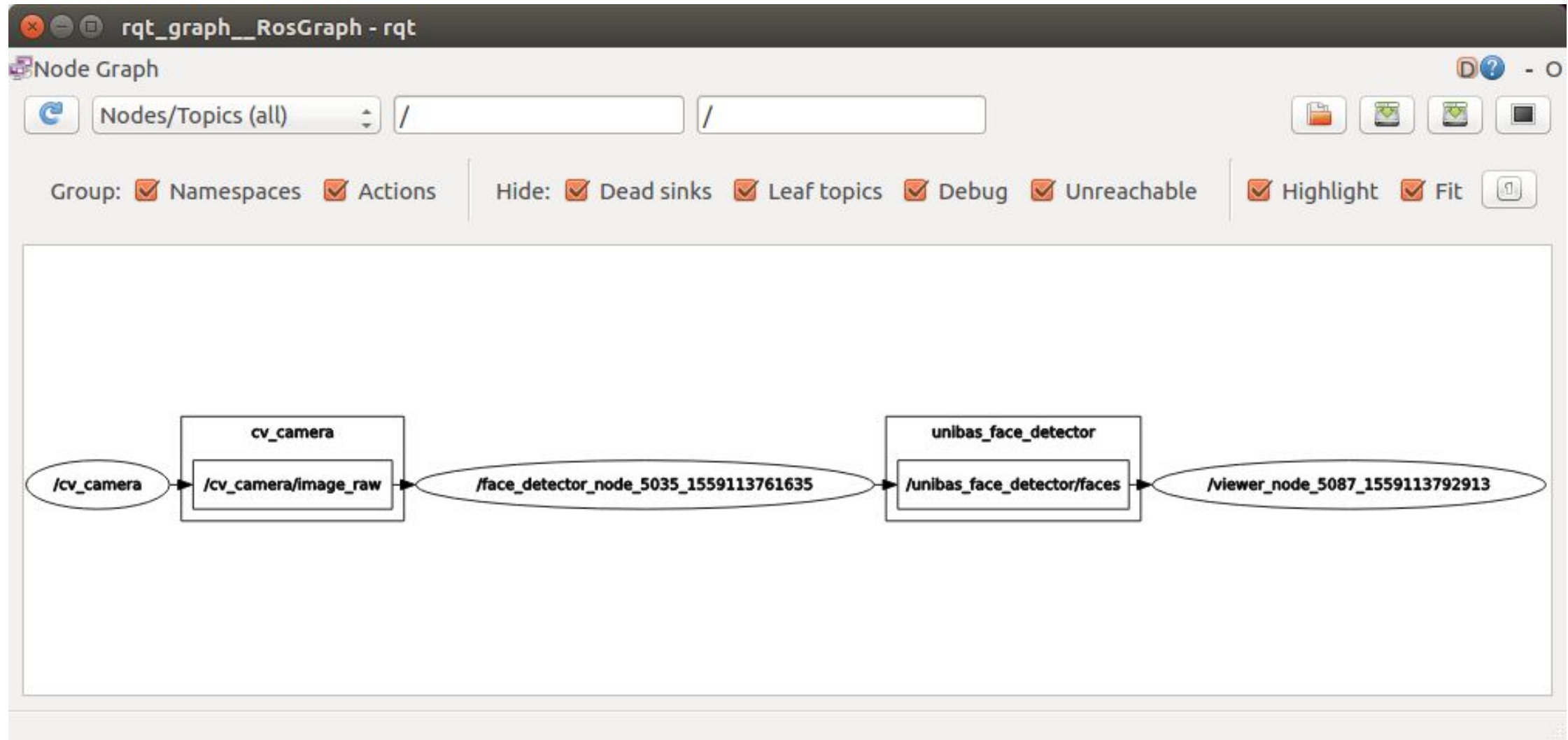
A terminal window with a dark purple background and a grey title bar. The title bar contains three window control icons (close, minimize, maximize) and the text 'bloisi@bloisi-U36SG: ~'. The terminal shows a command prompt where the user has entered 'rosrun unibas_face_detector face_detector_node.py'. The output of the command is '[INFO:0] Initialize OpenCL runtime...'.

```
bloisi@bloisi-U36SG: ~  
bloisi@bloisi-U36SG:~$ rosrun unibas_face_detector face_detector_node.py  
[ INFO:0] Initialize OpenCL runtime...
```

lanciamo il nodo viewer_node



rqt_graph



Esercizio 1

Utilizzare la rosbag people.bag

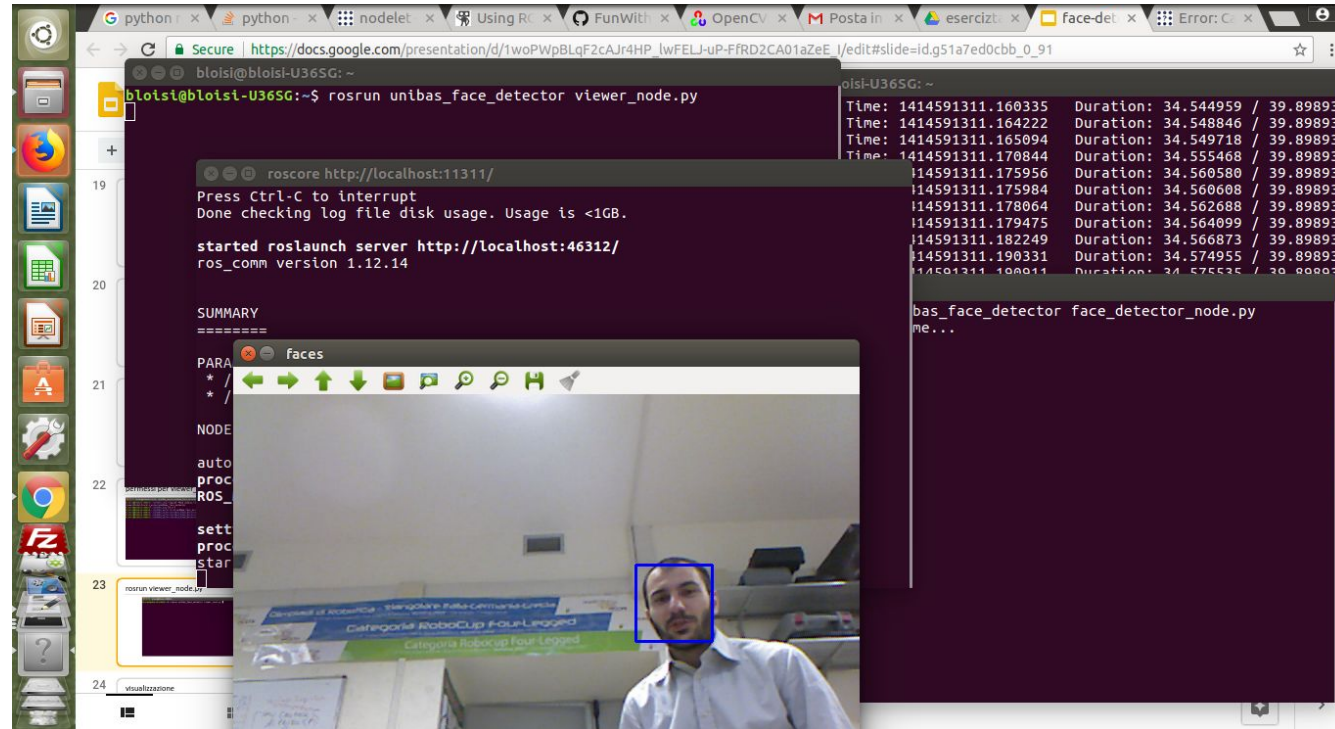
<https://drive.google.com/file/d/1oOMahlPdlwJkHMqXLtrLMktfx68-AGfJ/view?usp=sharing>

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



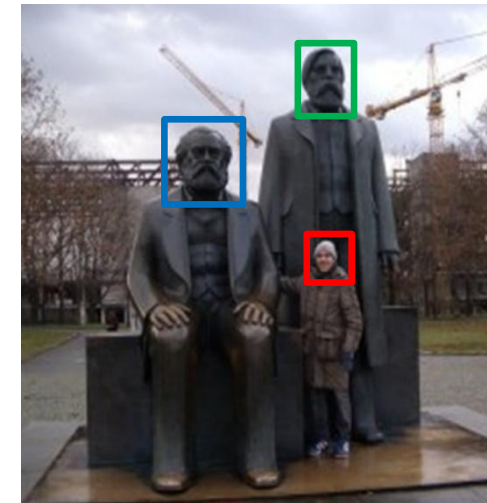
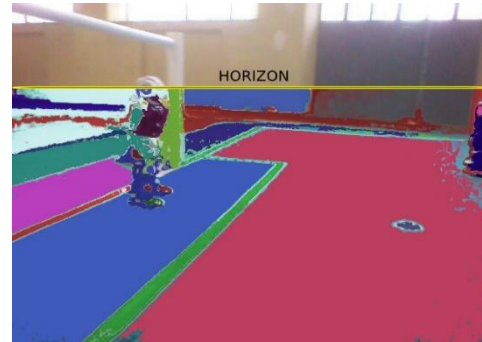
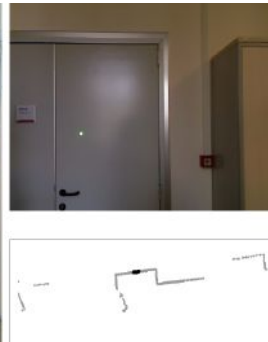
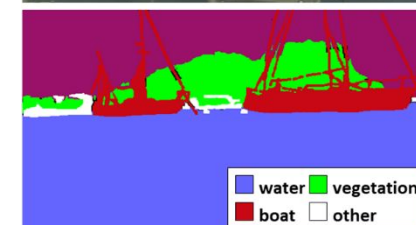


**UNIVERSITÀ DEGLI STUDI
DELLA BASILICATA**

Corso di Sistemi Informativi
A.A. 2018/19

Docente
Domenico Daniele Bloisi

face detection



Giugno 2019