



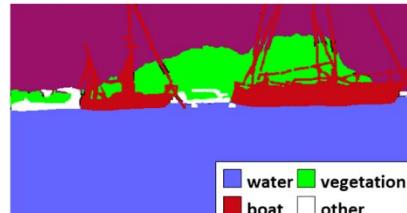
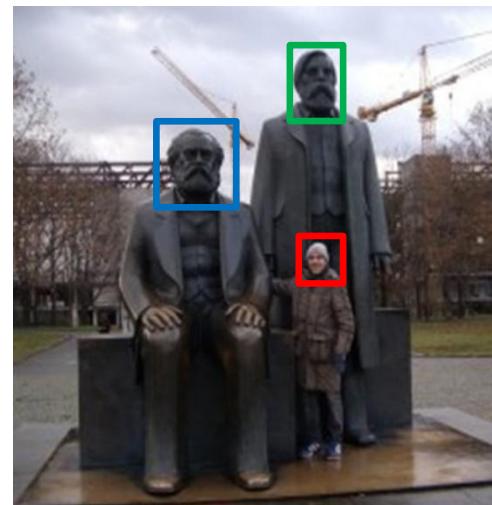
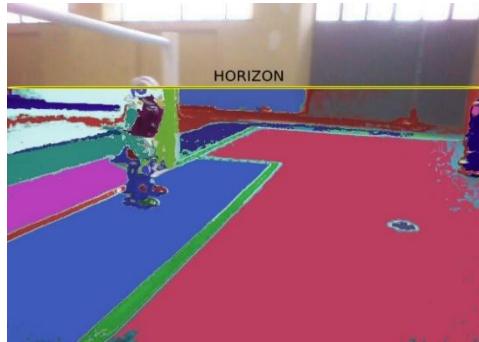
**UNIVERSITÀ DEGLI STUDI  
DELLA BASILICATA**

*Corso di Visione e Percezione*  
A.A. 2019/2020

Docente  
**Domenico Daniele Bloisi**

# Face detection in ROS

*Giugno 2020*



# obiettivo

---

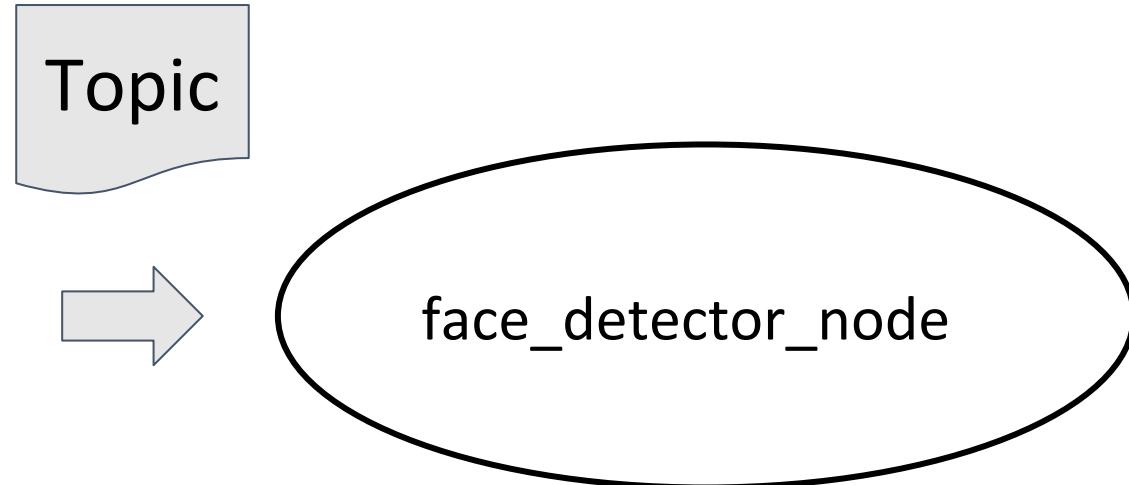
- Vogliamo realizzare un package ROS per la detection di volti
- Il package dovrà contenere due nodi:
  - il primo nodo servirà per rilevare i volti presenti nelle immagini a provenienti da una bag o da un sensore reale
  - il secondo nodo si occuperà di mostrare i risultati a video

# package unibas\_face\_detector

---

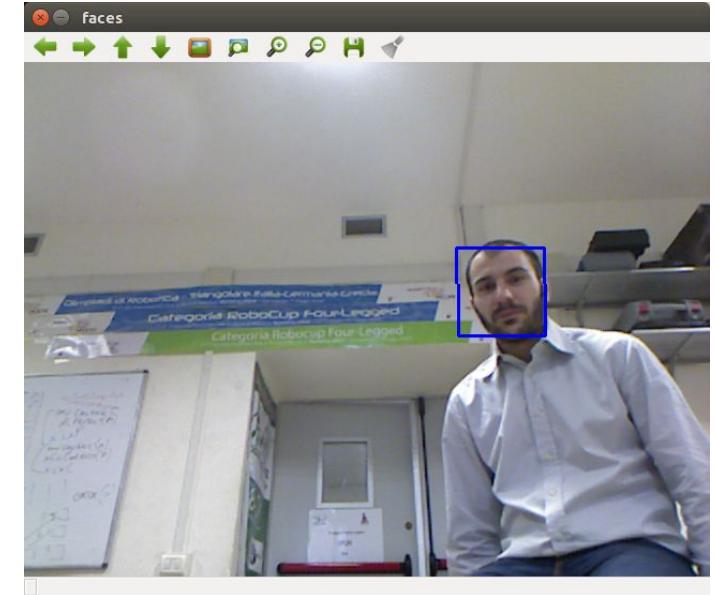
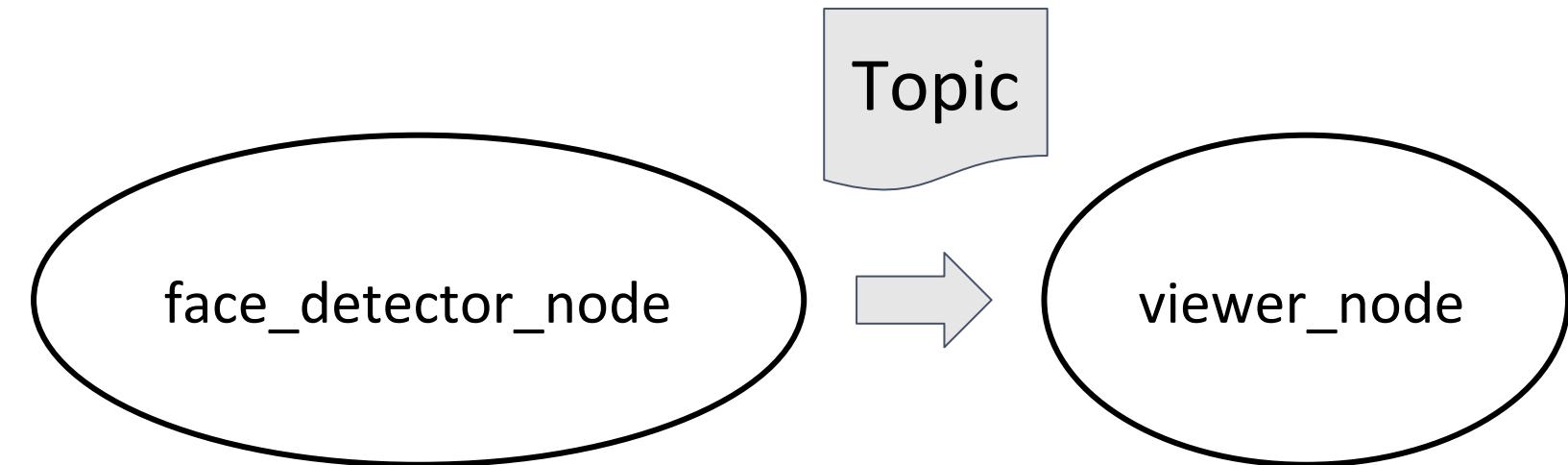


immagini live  
acquisite con una  
**telecamera**  
oppure una  
**rosbag**



# face\_detector\_node e viewer\_node

---



visualizzazione  
immagine OpenCV

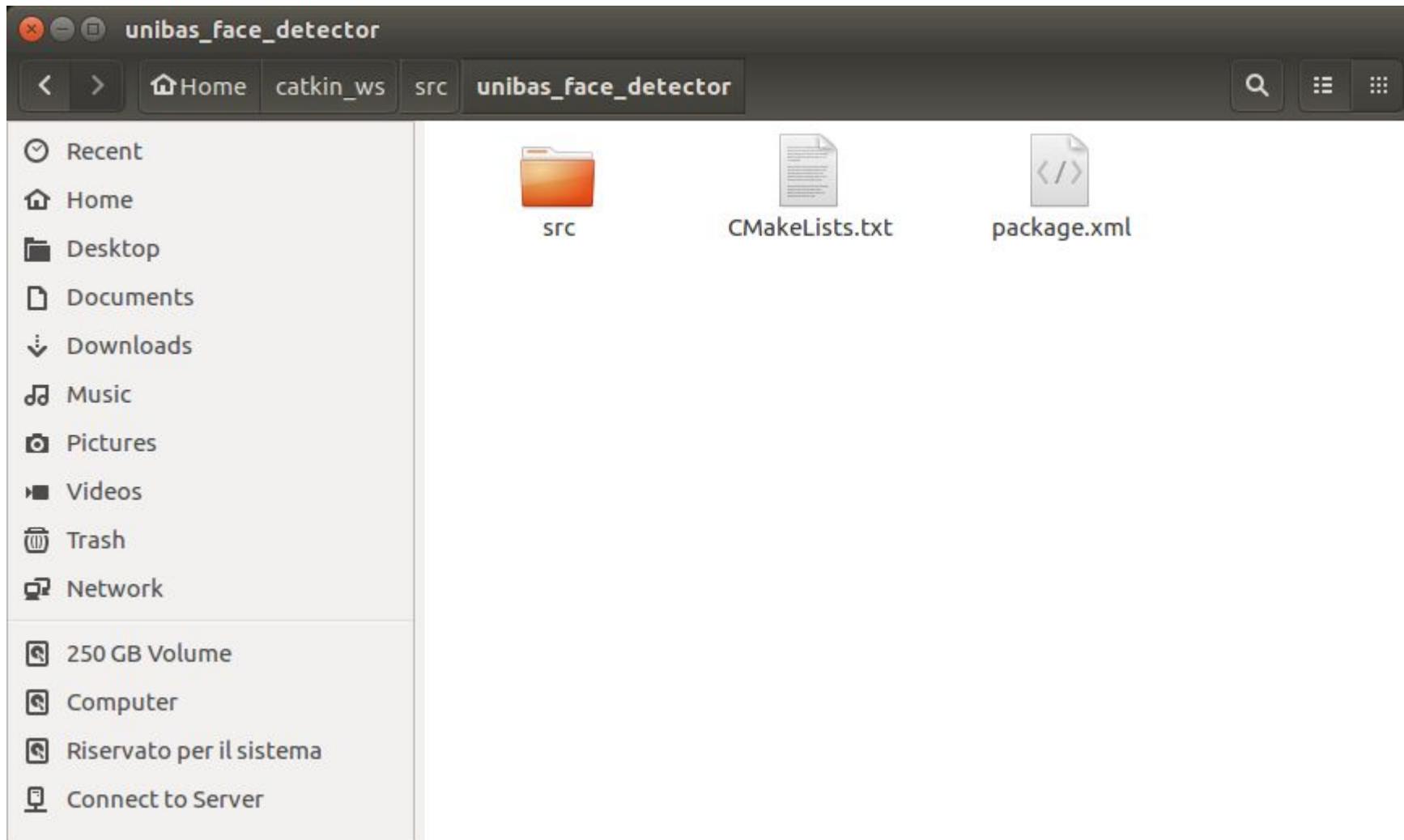
# creazione unibas\_face\_detector

---

```
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. Please 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:~/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

```
bloisi@bloisi-U36SG: ~/catkin_ws
[ 52%] Built target hw1_generate_messages_py
[ 53%] Built target hw1_generate_messages_lisp
. ~/catkin_ws/devel/setup.bash
[ 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

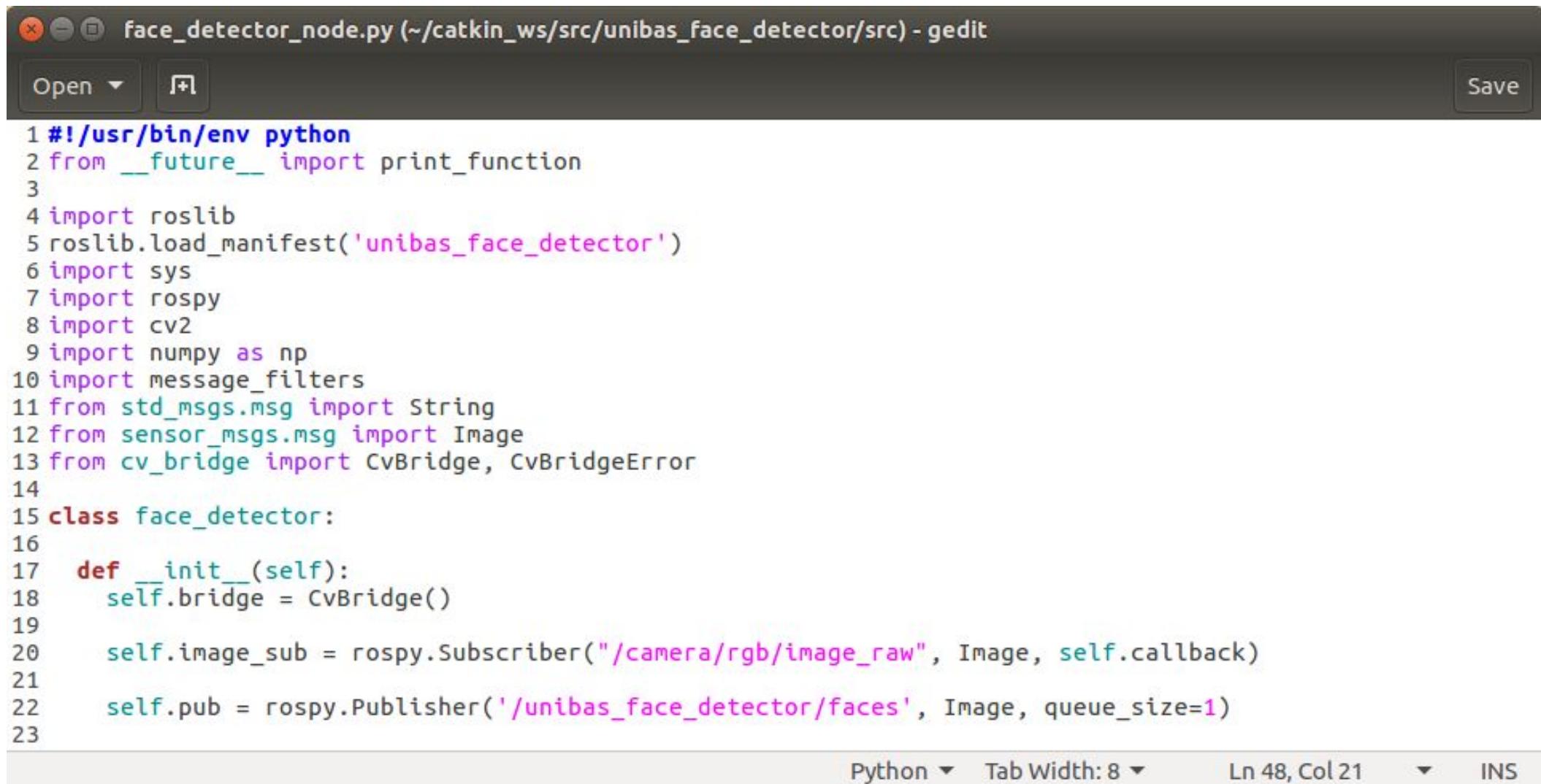
# creiamo face\_detector\_node.py

---



# codice face\_detector\_node.py

---

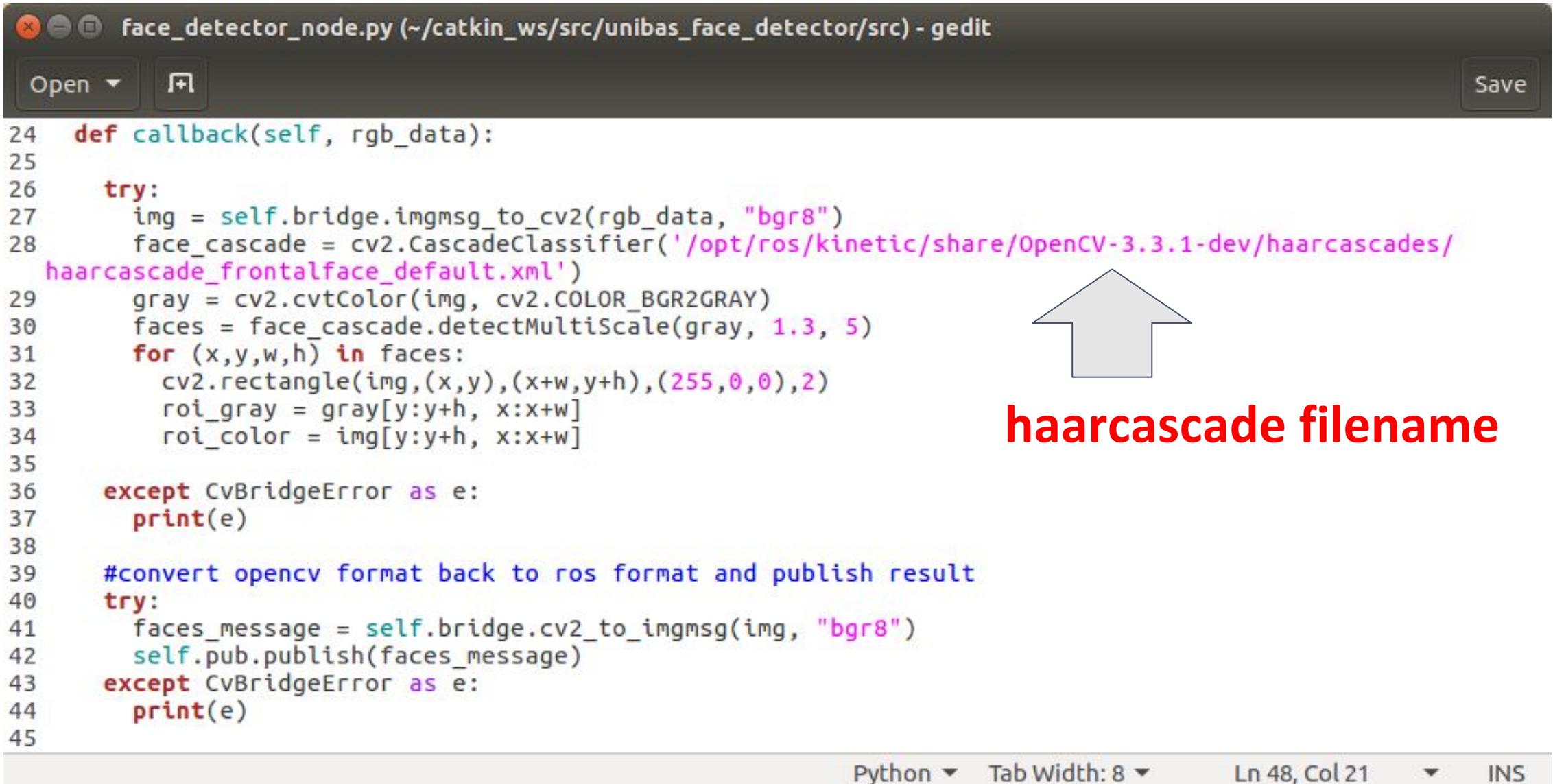


The screenshot shows a Gedit text editor window with the title "face\_detector\_node.py (~/catkin\_ws/src/unibas\_face\_detector/src) - gedit". The window contains Python code for a ROS node. The code imports various ROS and OpenCV modules and defines a class "face\_detector" with an \_\_init\_\_ method that subscribes to a camera topic and publishes detected faces. The code is numbered from 1 to 23.

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

haarcascade filename

Python ▾ Tab Width: 8 ▾ Ln 48, Col 21 ▾ INS

<https://github.com/opencv/opencv/tree/master/data/haarcascades>

The screenshot shows a GitHub repository page for the 'haarcascades' folder within the 'data' directory of the OpenCV master branch. The repository has 2.6k stars and 36.5k forks. The code tab is selected, showing a list of XML files related to Haar cascades for various objects like eyes, faces, and bodies. Most files were last updated 7 years ago, except for a few from 2 months ago.

File	Description	Last Updated
haarcascade_eye.xml	some attempts to tune the performance	7 years ago
haarcascade_eye_tree_eyeglasses.xml	some attempts to tune the performance	7 years ago
haarcascade_frontalcatface.xml	fix files permissions	2 months ago
haarcascade_frontalcatface_extend... haarcascade_frontalface_alt.xml	fix files permissions some attempts to tune the performance	2 months ago 7 years ago
haarcascade_frontalface_alt2.xml	some attempts to tune the performance	7 years ago
haarcascade_frontalface_alt_tree.xml	some attempts to tune the performance	7 years ago
haarcascade_frontalface_default.xml	some attempts to tune the performance	7 years ago
haarcascade_fullbody.xml	Some mist. typo fixes	2 years ago
haarcascade_lefteye_2splits.xml	some attempts to tune the performance	7 years ago

# codice face\_detector\_node.py

---

The screenshot shows a Gedit window displaying a Python script named `face_detector_node.py`. The window title bar reads "face\_detector\_node.py (~catkin\_ws/src/unibas\_face\_detector/src) - gedit". The toolbar includes "Open", "Save", and other standard file operations. The code itself is a ROS node implementation:

```
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
```

The status bar at the bottom indicates the language is "Python", tab width is 8, the current line is 48, column 21, and there is an "INS" indicator.

# 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
* /rosdistro: kinetic
* /rosversion: 1.12.14

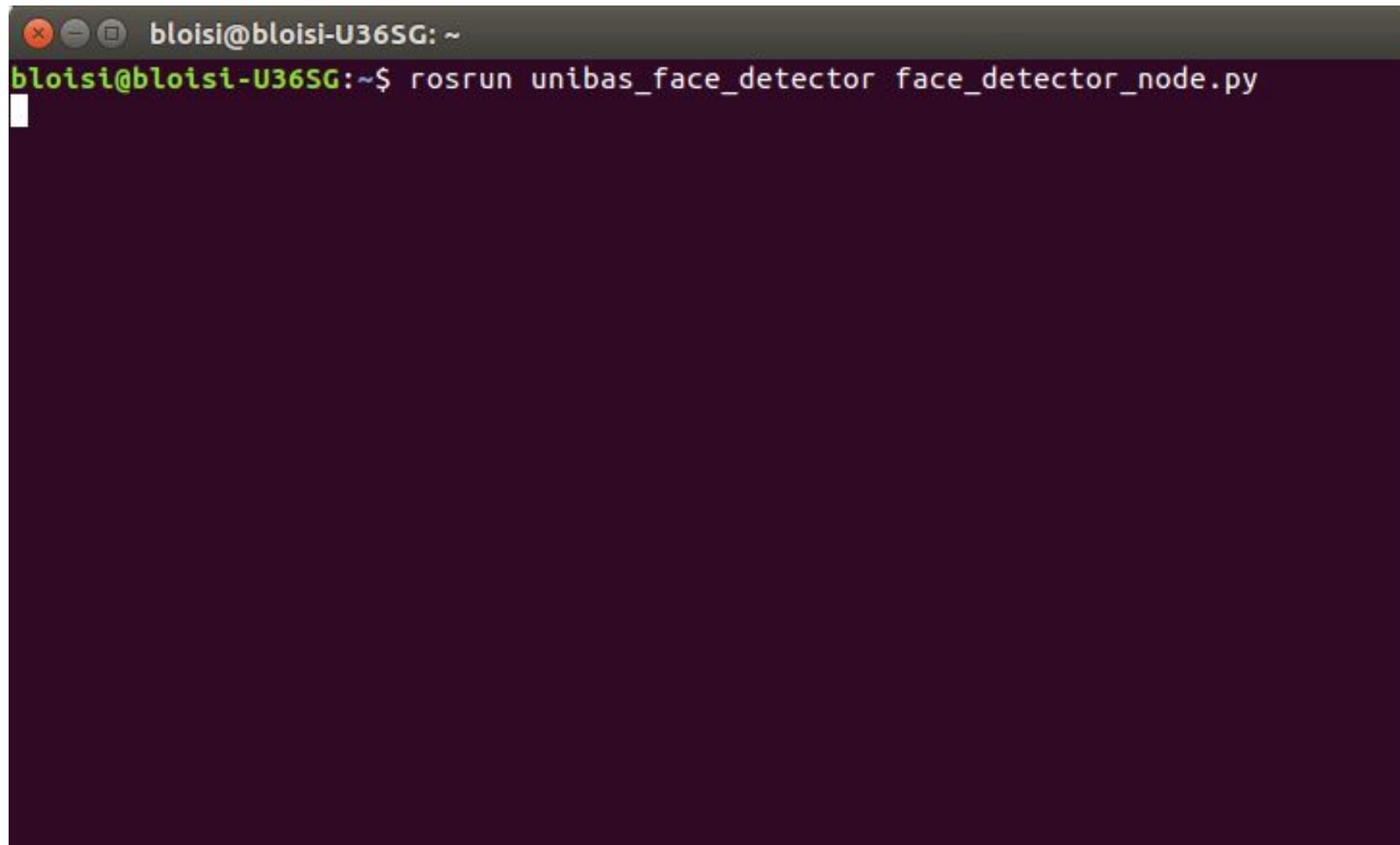
NODES

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

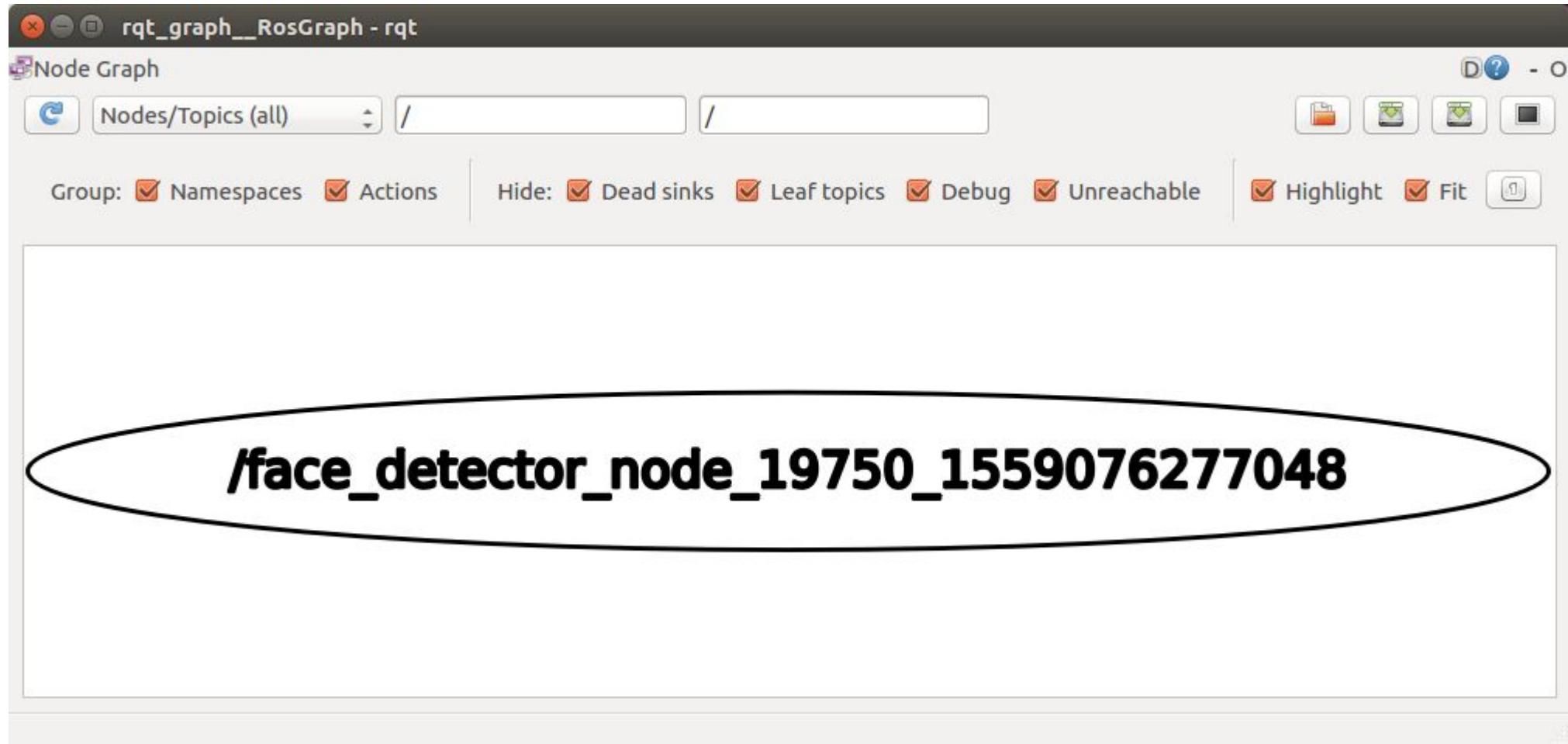
---

A screenshot of a terminal window titled "bloisi@bloisi-U36SG: ~". The window contains a single line of text: "bloisi@bloisi-U36SG:~\$ rosrun unibas\_face\_detector face\_detector\_node.py". The terminal has a dark background and light-colored text. The title bar is at the top, and the command is entered in the main body of the window.

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

# rqt\_graph

---



# rosbag play

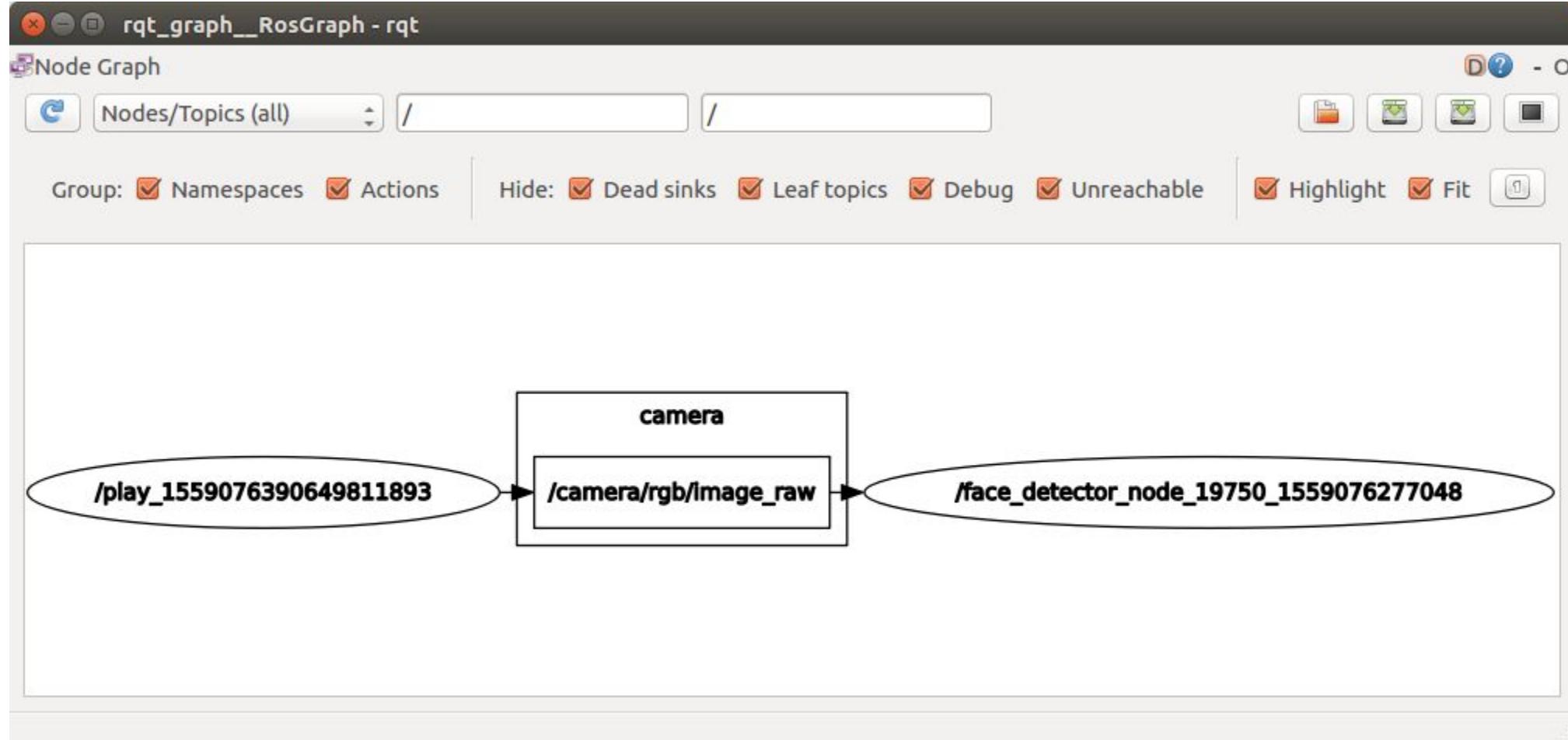
---

```
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.

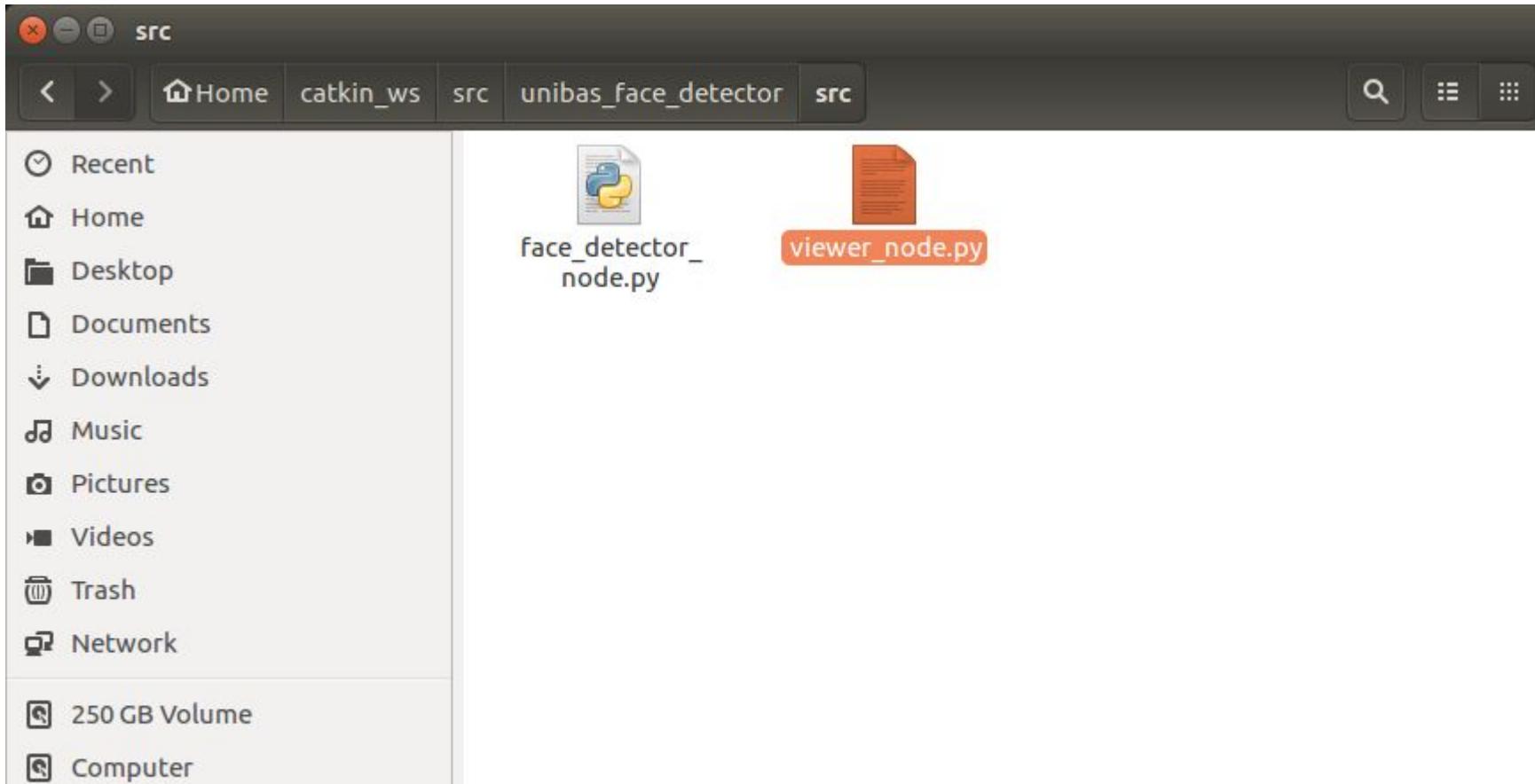
Hit space to toggle paused, or 's' to step.
[RUNNING] Bag Time: 1414591276.615376 Duration: 0.000000 / 39.898938
[RUNNING] Bag Time: 1414591276.784976 Duration: 0.169601 / 39.898938
[RUNNING] Bag Time: 1414591276.802653 Duration: 0.187277 / 39.898938
[RUNNING] Bag Time: 1414591276.804009 Duration: 0.188634 / 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



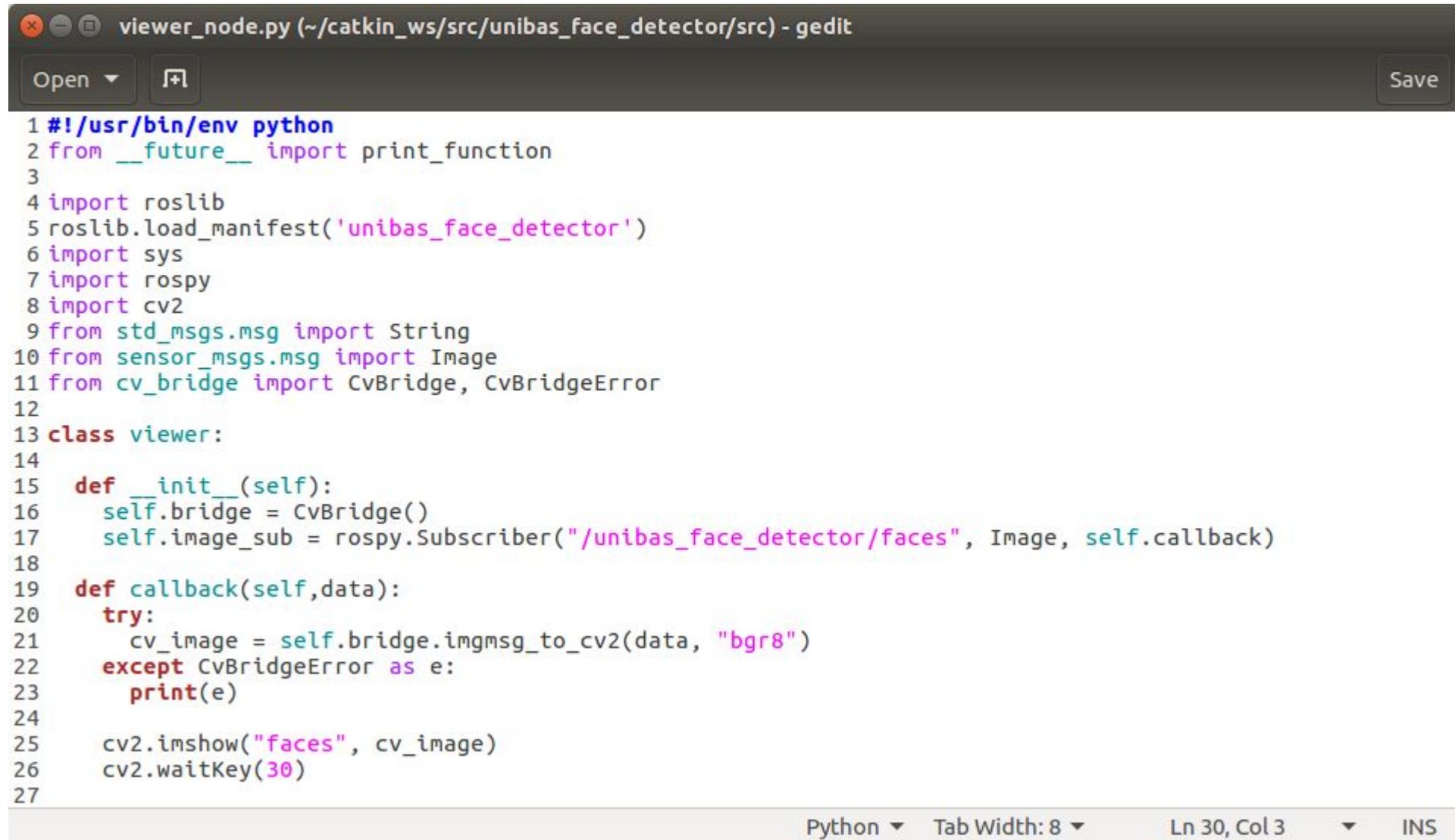
# viewer\_node

---



# codice viewer\_node

---



The screenshot shows a Gedit text editor window titled "viewer\_node.py (~/catkin\_ws/src/unibas\_face\_detector/src) - gedit". The code is a Python script for a ROS node named "viewer". It imports necessary modules from roslib, sys, rospy, cv2, std\_msgs.msg, sensor\_msgs.msg, and cv\_bridge. The "viewer" class has an \_\_init\_\_ method that initializes a CvBridge and a Subscriber to the "/unibas\_face\_detector/faces" topic. The callback function converts the received ROS message to a CV image, handles CvBridge errors, and displays the image using cv2.imshow. The code ends with cv2.waitKey(30). The status bar at the bottom indicates the file is saved in Python mode with a tab width of 8, and the current line is 30, column 3.

```
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

---



The screenshot shows a terminal window with the following text output:

```
[INFO] [WallclockTime: 1581133311.118000] [unibas_face_detector]: Starting up.
```

This indicates that the ROS node has been successfully launched.

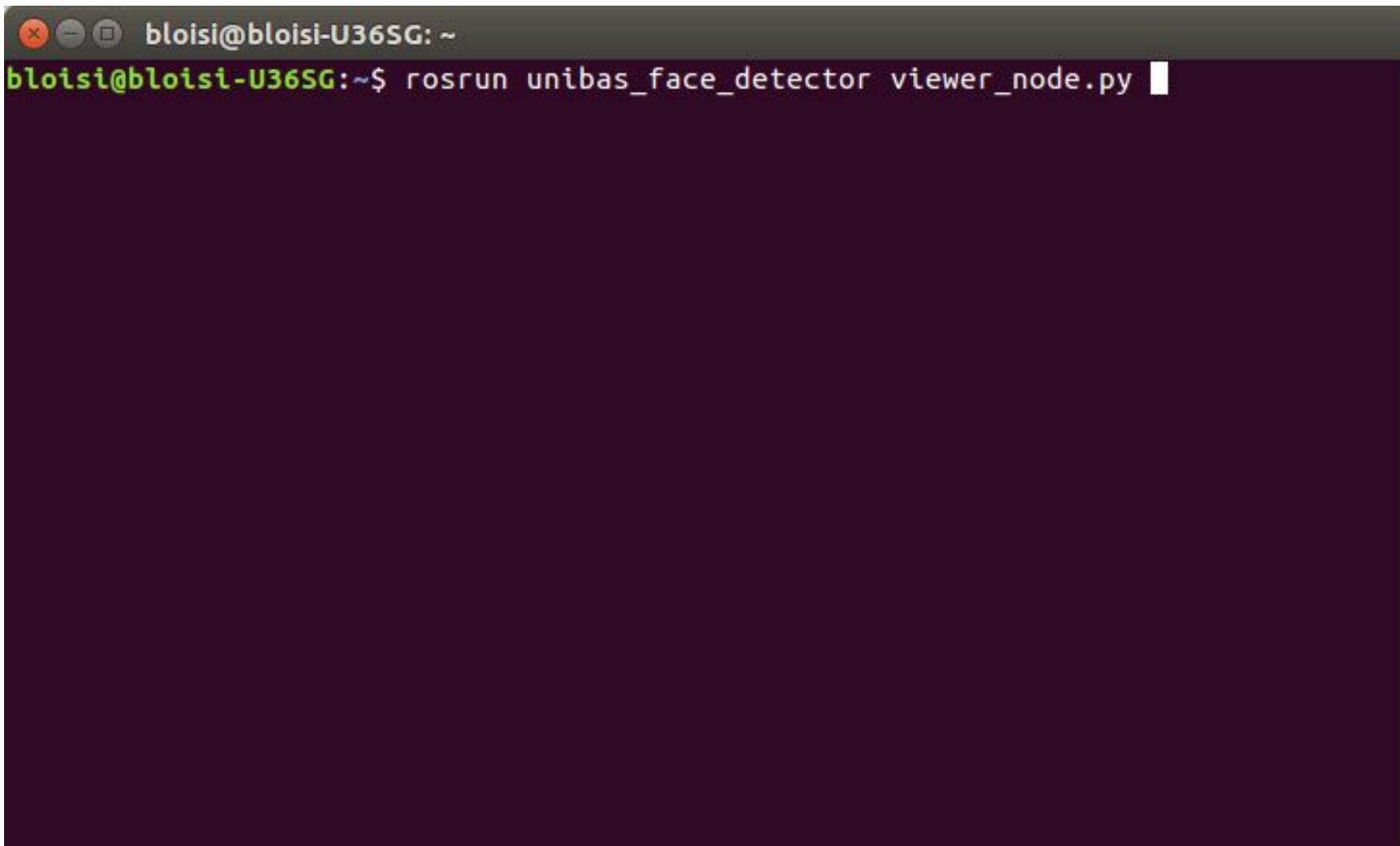
# permessi per viewer\_node.py

---

```
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$ █
```

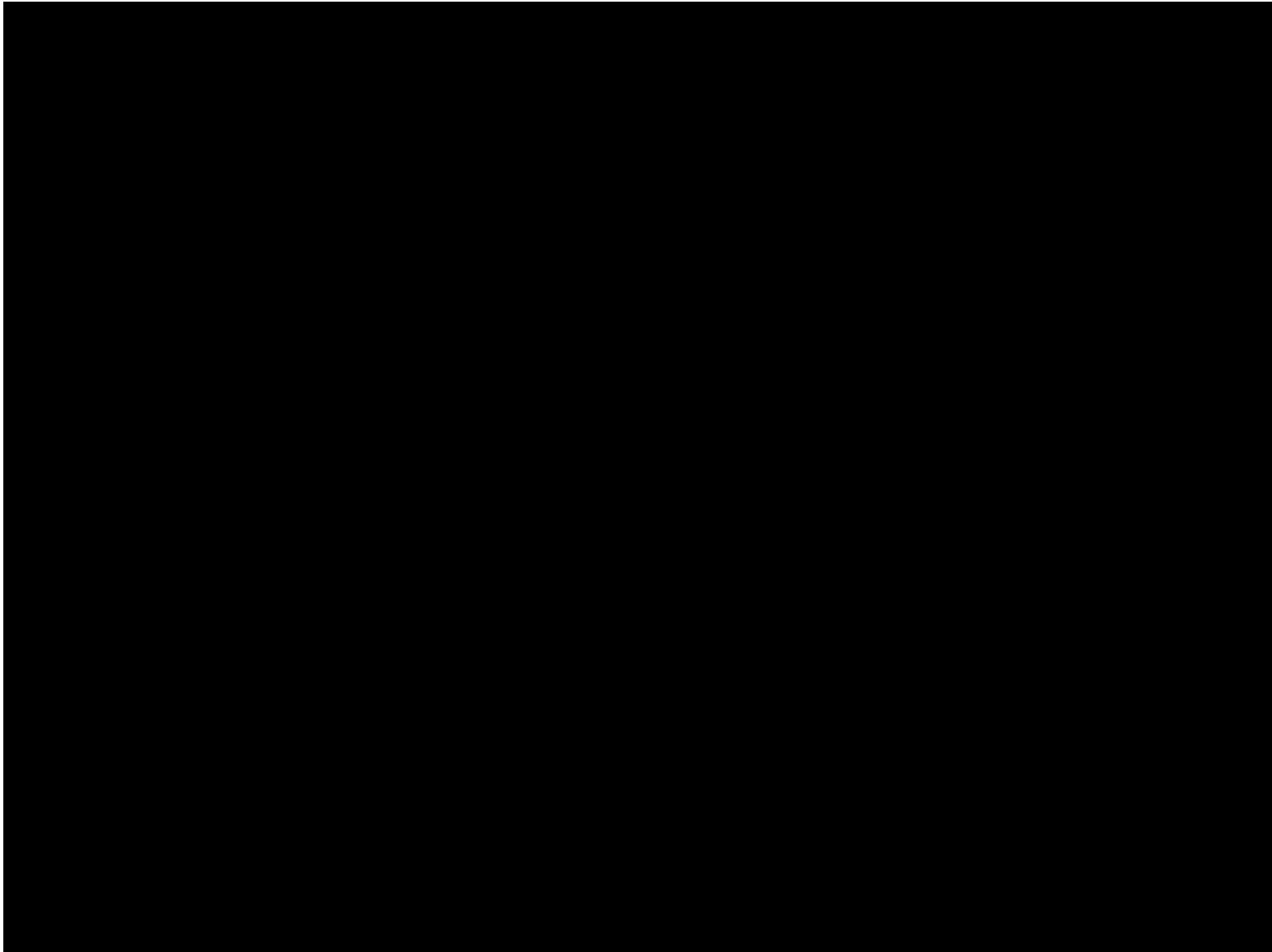
# rosrun viewer\_node.py

---

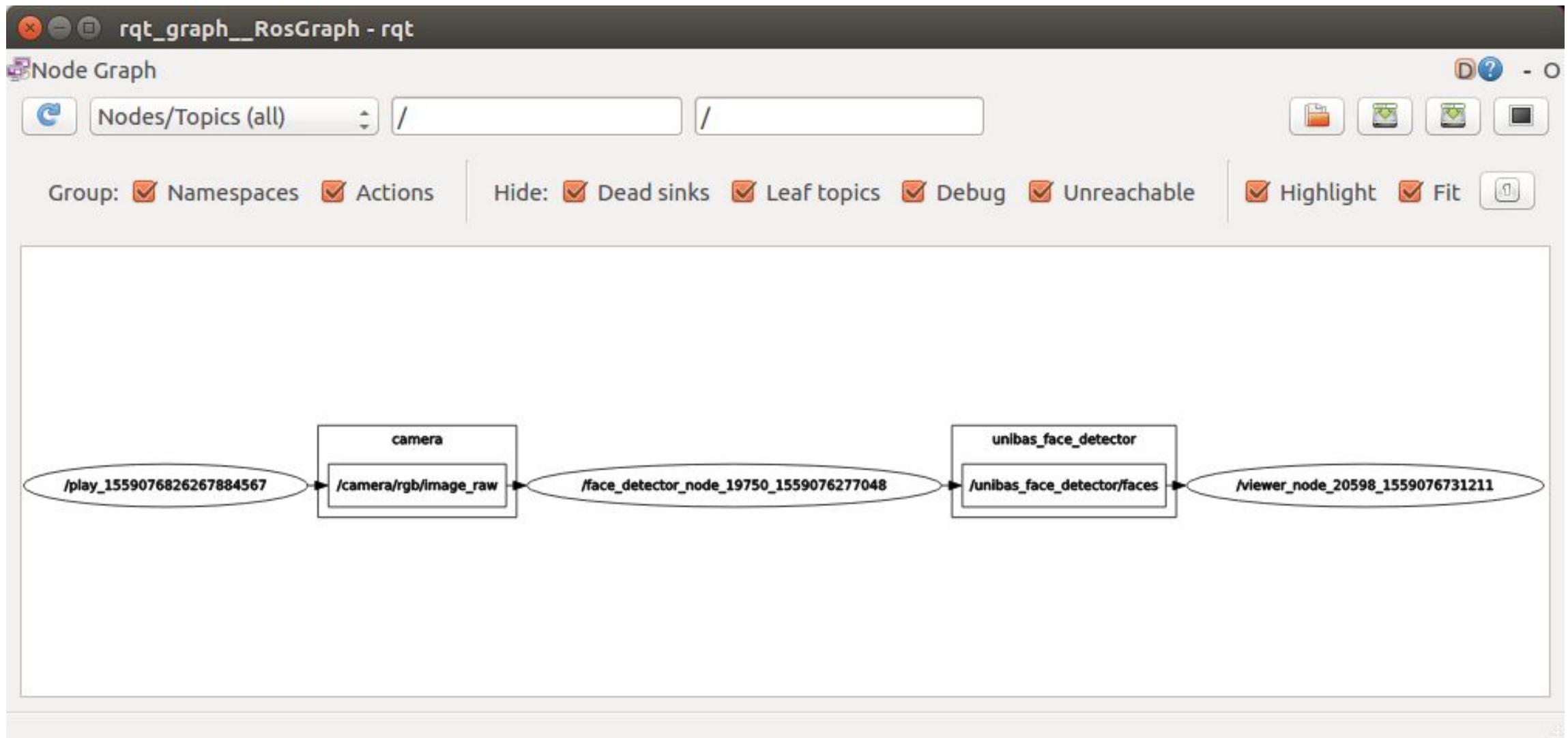
A screenshot of a terminal window titled "bloisi@bloisi-U36SG: ~". The window contains a single line of text: "bloisi@bloisi-U36SG:~\$ rosrun unibas\_face\_detector viewer\_node.py". The terminal has a dark background with light-colored text and standard window controls at the top.

# visualizzazione

---



# rqt\_graph

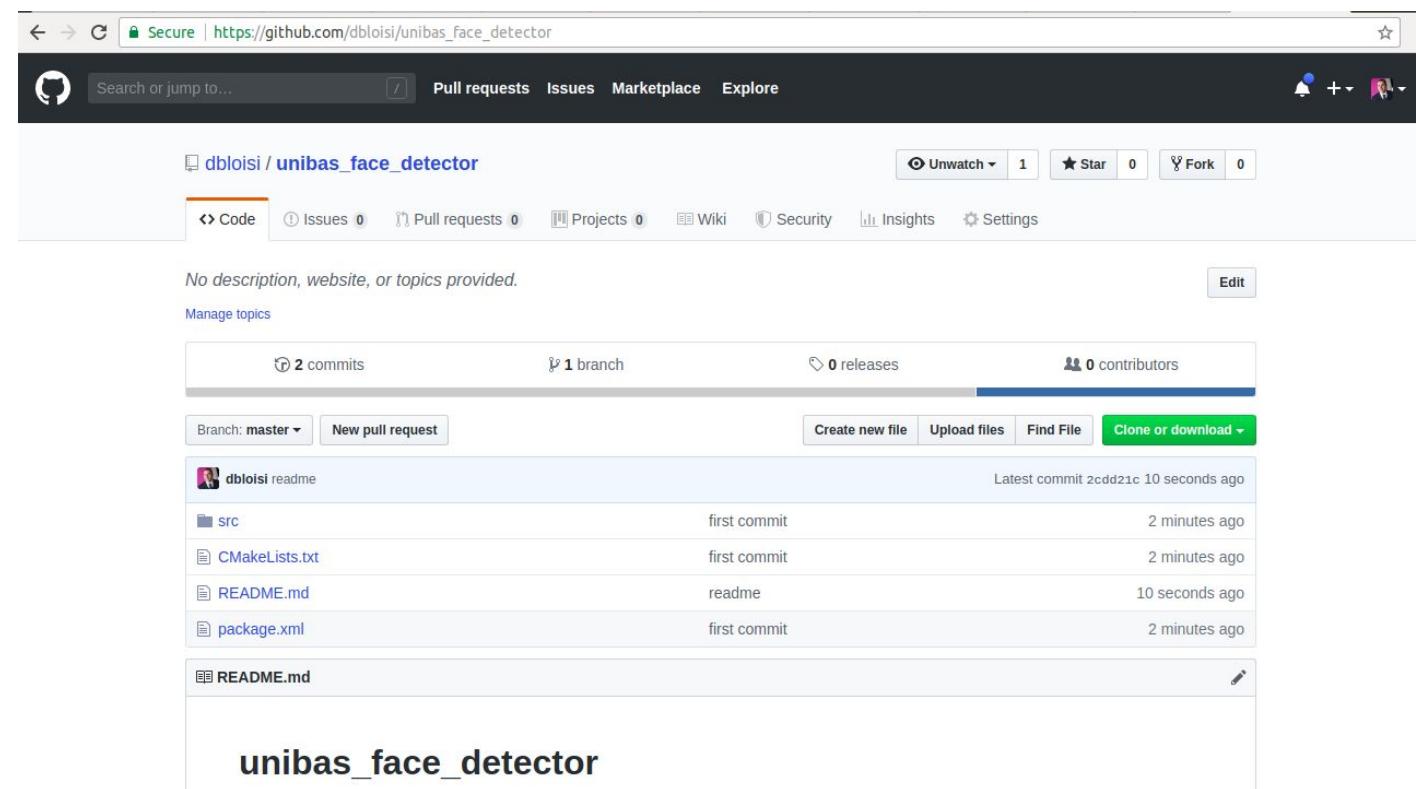


# repository unibas\_face\_detector

---

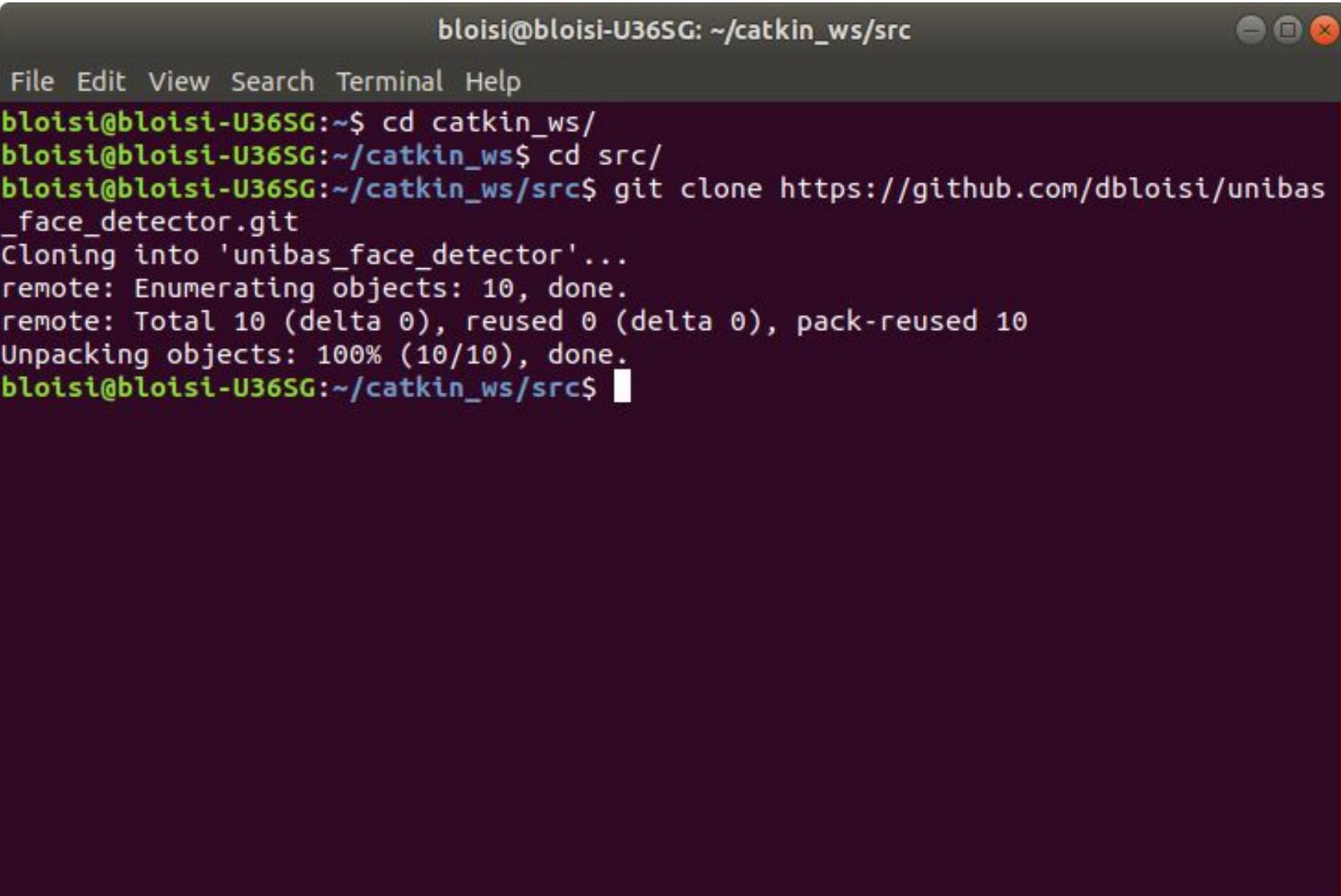
Il codice del repository `unibas_face_detector` è disponibile al seguente link

[https://github.com/dbloisi/unibas\\_face\\_detector](https://github.com/dbloisi/unibas_face_detector)



# repository unibas\_face\_detector

---



A screenshot of a terminal window titled "bloisi@bloisi-U36SG: ~/catkin\_ws/src". The window has a dark theme with orange window controls. The terminal menu bar includes "File", "Edit", "View", "Search", "Terminal", and "Help". The command-line session shows the user navigating to their workspace and cloning a GitHub repository:

```
bloisi@bloisi-U36SG:~$ cd catkin_ws/
bloisi@bloisi-U36SG:~/catkin_ws$ cd src/
bloisi@bloisi-U36SG:~/catkin_ws/src$ git clone https://github.com/dbloisi/unibas_face_detector.git
Cloning into 'unibas_face_detector'...
remote: Enumerating objects: 10, done.
remote: Total 10 (delta 0), reused 0 (delta 0), pack-reused 10
Unpacking objects: 100% (10/10), done.
bloisi@bloisi-U36SG:~/catkin_ws/src$
```

# repository unibas\_face\_detector

---

```
bloisi@bloisi-U36SG: ~/catkin_ws
File Edit View Search Terminal Help
bloisi@bloisi-U36SG:~/catkin_ws/src$ cd ..
bloisi@bloisi-U36SG:~/catkin_ws$ catkin_make --pkg unibas_face_detector
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"
#####
CMake Warning (dev) in CMakeLists.txt:
  No project() command is present.  The top-level CMakeLists.txt file must
  contain a literal, direct call to the project() command.  Add a line of
  code such as

    project(ProjectName)

  near the top of the file, but after cmake_minimum_required().
  CMake is pretending there is a "project(Project)" command on the first
  line.
This warning is for project developers.  Use -Wno-dev to suppress it.
```

# repository unibas\_face\_detector

---

```
bloisi@bloisi-U36SG: ~/catkin_ws
File Edit View Search Terminal Help
-- +++ processing catkin package: 'turtlebot3_teleop'
-- ==> add_subdirectory(turtlebot3/turtlebot3_teleop)
-- +++ processing catkin package: 'unibas_face_detector'
-- ==> add_subdirectory(unibas_face_detector)
-- +++ processing catkin package: 'unibas_teleop'
-- ==> add_subdirectory(unibas_teleop)
-- +++ processing catkin package: 'unibas_turtle'
-- ==> add_subdirectory(unibas_turtle)
-- +++ processing catkin package: 'unibas_viewer'
-- ==> add_subdirectory(unibas_viewer)
-- +++ processing catkin package: 'turtlebot3_description'
-- ==> add_subdirectory(turtlebot3/turtlebot3_description)
-- +++ processing catkin package: 'yolo_ros_detect'
-- ==> add_subdirectory(yolo_ros_detect)
-- Using these message generators: gencpp;geneus;genlisp;gennodejs;genpy
-- Found Boost: /usr/include (found version "1.65.1") found components: system
-- Configuring done
-- Generating done
-- Build files have been written to: /home/bloisi/catkin_ws/build
#####
##### Running command: "make -j4 -l4" in "/home/bloisi/catkin_ws/build/unibas_fac
e_detector"
#####
bloisi@bloisi-U36SG:~/catkin_ws$
```

# input da webcam

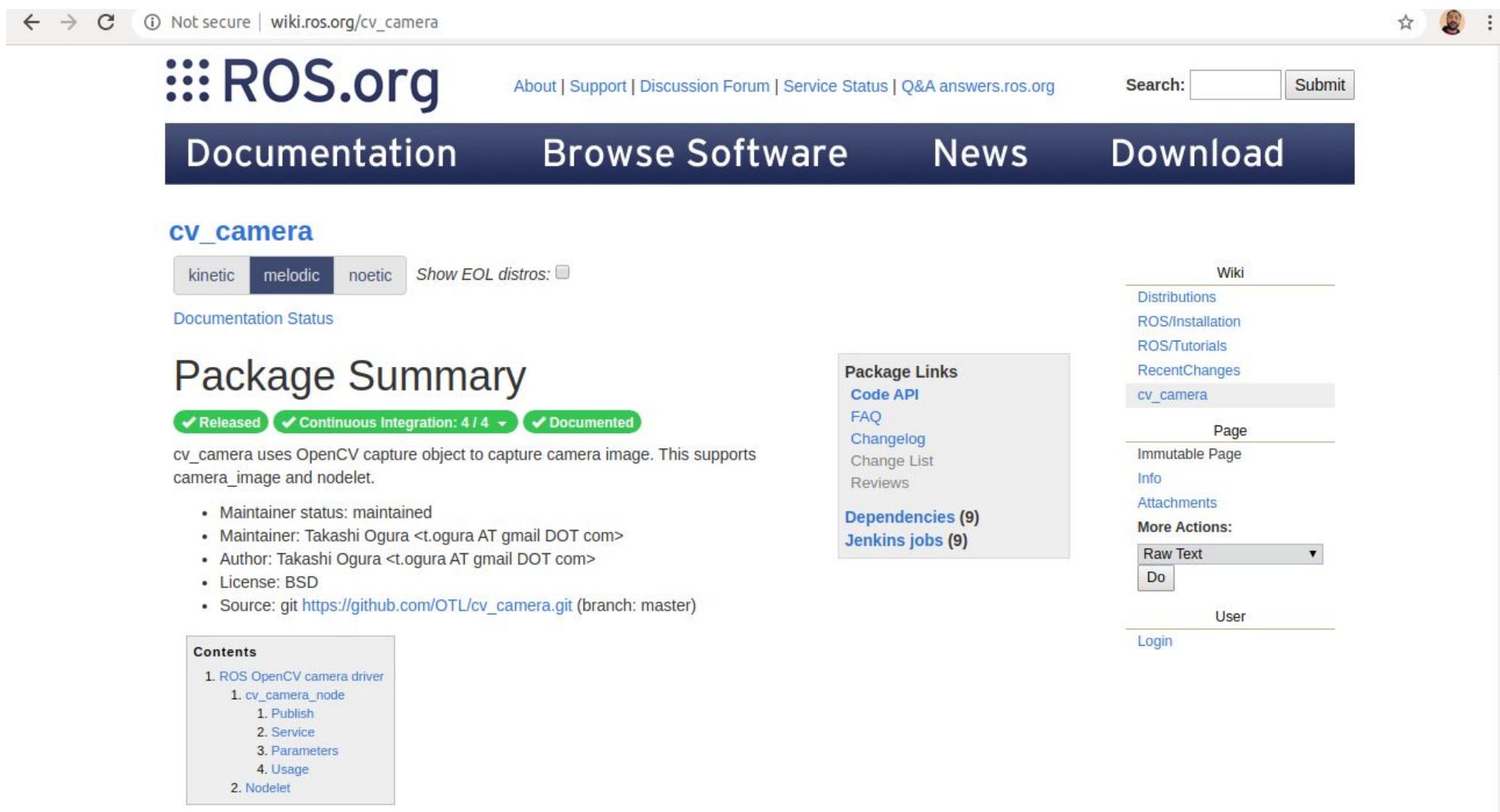
---

Proviamo ad utilizzare le immagini provenienti dalla webcam del nostro pc

Ci servirà un nodo per

1. leggere lo stream della webcam
2. pubblicarlo su un topic ROS

# package cv\_camera



The screenshot shows the ROS.org website with the URL [wiki.ros.org/cv\\_camera](https://wiki.ros.org/cv_camera). The page title is "cv\_camera". The top navigation bar includes links for Documentation, Browse Software, News, and Download. On the left, there's a sidebar with "cv\_camera" and tabs for kinetic, melodic, and noetic. Below that is a "Documentation Status" section with green checkmarks for Released, Continuous Integration (4/4), and Documented. The main content area features a "Package Summary" section with a brief description and a bulleted list of maintainer information. To the right is a "Package Links" sidebar with links to Code API, FAQ, Changelog, Change List, and Reviews. Another sidebar lists Dependencies (9) and Jenkins jobs (9). On the far right, there are sections for Wiki (Distributions, ROS/Installation, ROS/Tutorials, RecentChanges, cv\_camera), Page (Immutable Page, Info, Attachments), More Actions (Raw Text, Do), and User (Login).

cv\_camera

kinetic melodic noetic Show EOL distros:

Documentation Status

## Package Summary

✓ Released ✓ Continuous Integration: 4 / 4 ✓ Documented

cv\_camera uses OpenCV capture object to capture camera image. This supports camera\_image and nodelet.

- Maintainer status: maintained
- 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](https://github.com/OTL/cv_camera.git) (branch: master)

**Contents**

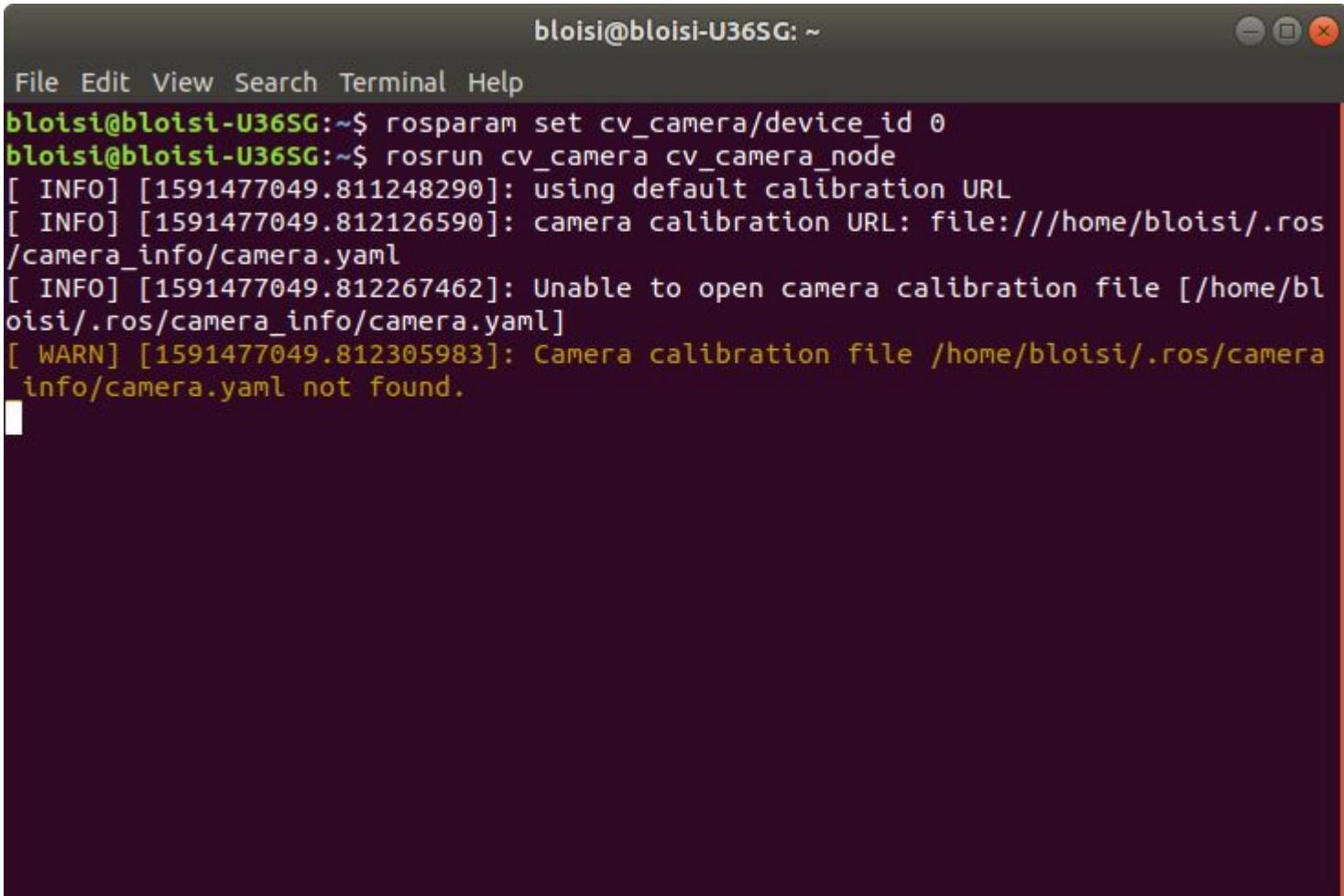
- ROS OpenCV camera driver
  - cv\_camera\_node
    - Publish
    - Service
    - Parameters
    - Usage
  - Nodelet

# installazione package cv\_camera

```
bloisi@bloisi-U36SG: ~
File Edit View Search Terminal Help
bloisi@bloisi-U36SG:~$ sudo apt-get install ros-melodic-cv-camera
Reading package lists... Done
Building dependency tree
Reading state information... Done
sudo apt-get install ros-melodic-cv-camera
Use 'sudo apt autoremove' to remove it.
The following NEW packages will be installed:
  ros-melodic-cv-camera
0 upgraded, 1 newly installed, 0 to remove and 202 not upgraded.
Need to get 55,8 kB of archives.
After this operation, 249 kB of additional disk space will be used.
Get:1 http://packages.ros.org/ros/ubuntu bionic/main amd64 ros-melodic-cv-camera
  amd64 0.4.0-1bionic.20200514.223237 [55,8 kB]
Fetched 55,8 kB in 1s (69,3 kB/s)
Selecting previously unselected package ros-melodic-cv-camera.
(Reading database ... 352548 files and directories currently installed.)
Preparing to unpack .../ros-melodic-cv-camera_0.4.0-1bionic.20200514.223237_amd6
4.deb ...
Unpacking ros-melodic-cv-camera (0.4.0-1bionic.20200514.223237) ...
Setting up ros-melodic-cv-camera (0.4.0-1bionic.20200514.223237) ...
bloisi@bloisi-U36SG:~$
```

# lanciamo cv\_camera

---

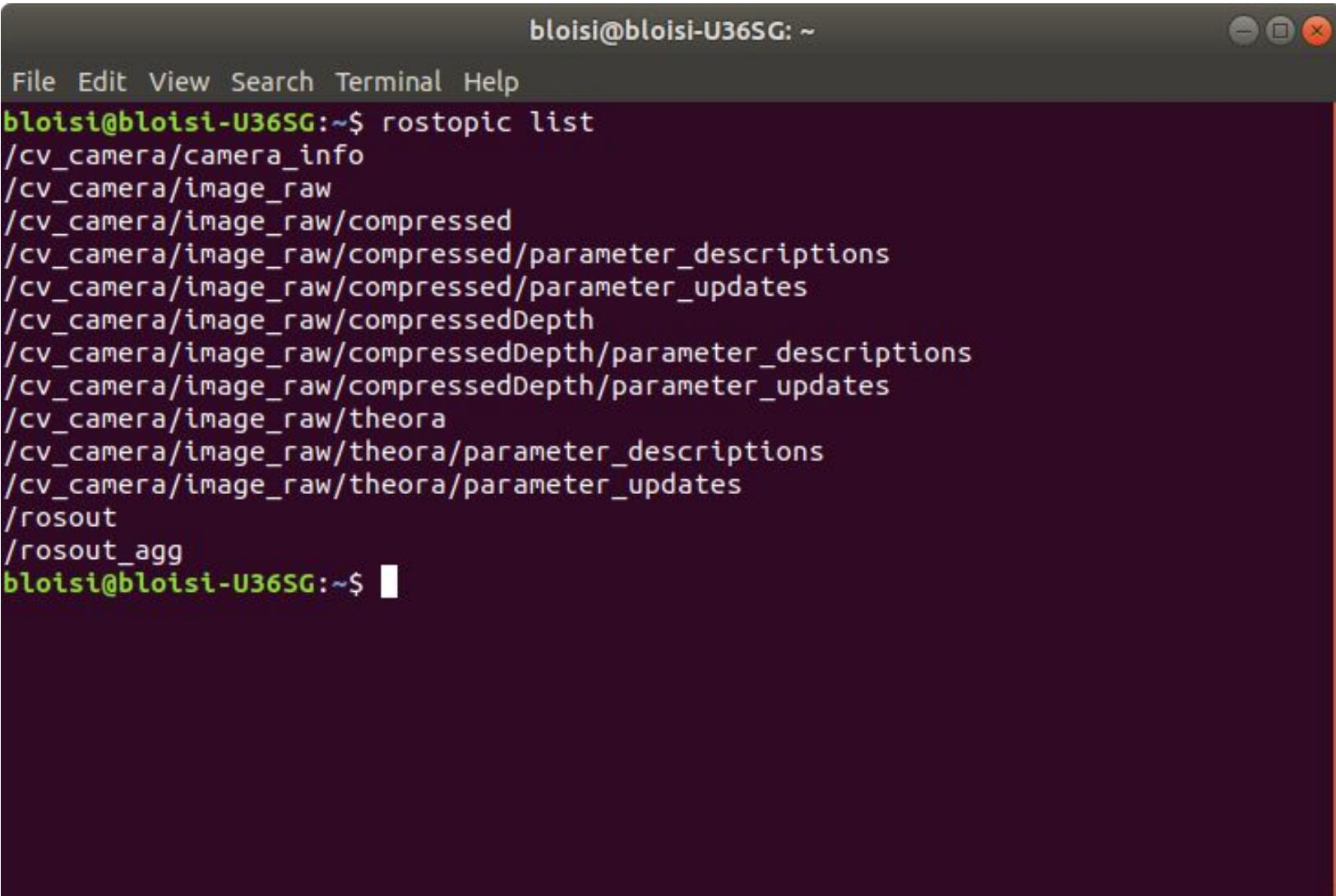


A screenshot of a terminal window titled "bloisi@bloisi-U36SG: ~". The window has a dark theme with orange window controls. The terminal menu bar includes "File", "Edit", "View", "Search", "Terminal", and "Help". The main pane displays the following ROS command execution:

```
bloisi@bloisi-U36SG:~$ rosparam set cv_camera/device_id 0
bloisi@bloisi-U36SG:~$ rosrun cv_camera cv_camera_node
[ INFO] [1591477049.811248290]: using default calibration URL
[ INFO] [1591477049.812126590]: camera calibration URL: file:///home/bloisi/.ros/camera_info/camera.yaml
[ INFO] [1591477049.812267462]: Unable to open camera calibration file [/home/bloisi/.ros/camera_info/camera.yaml]
[ WARN] [1591477049.812305983]: Camera calibration file /home/bloisi/.ros/camera_info/camera.yaml not found.
```

# topic di cv\_camera

---



A screenshot of a terminal window titled "bloisi@bloisi-U36SG: ~". The window has a dark background and light-colored text. At the top, there's a menu bar with "File", "Edit", "View", "Search", "Terminal", and "Help". Below the menu, the command "rostopic list" is run, followed by its output. The output shows various ROS topics starting with "/cv\_camera/".

```
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

---

```
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("/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
```

The screenshot shows a Gedit text editor window with the file "face\_detector\_node.py" open. The code is written in Python and defines a class "face\_detector". It includes imports for std\_msgs, sensor\_msgs, CvBridge, and CvBridgeError. The class has a constructor "\_\_init\_\_" that initializes a CvBridge object and subscribes to two topics: "/cv\_camera/image\_raw" and "/camera/rgb/image\_raw", both using the "self.callback" function as the callback. It also publishes to the topic "/unibas\_face\_detector/faces". A red circle highlights the line "self.image\_sub = rospy.Subscriber('/cv\_camera/image\_raw', Image, self.callback)". The status bar at the bottom indicates the code is in Python mode, the tab width is 8, and the current line is 20, column 51. There is also an "INS" indicator.

# lanciamo il nodo face\_detector\_node

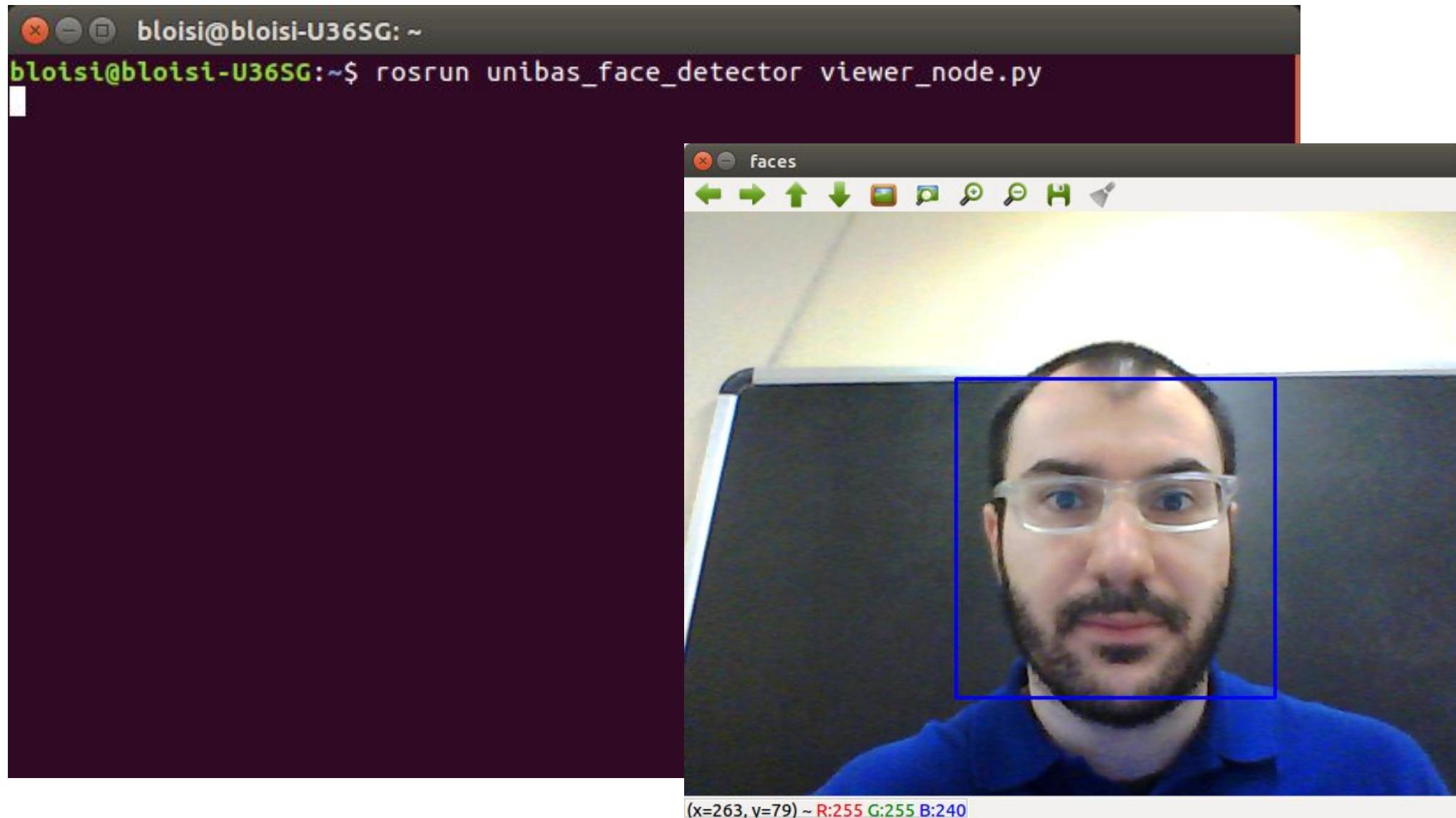
---



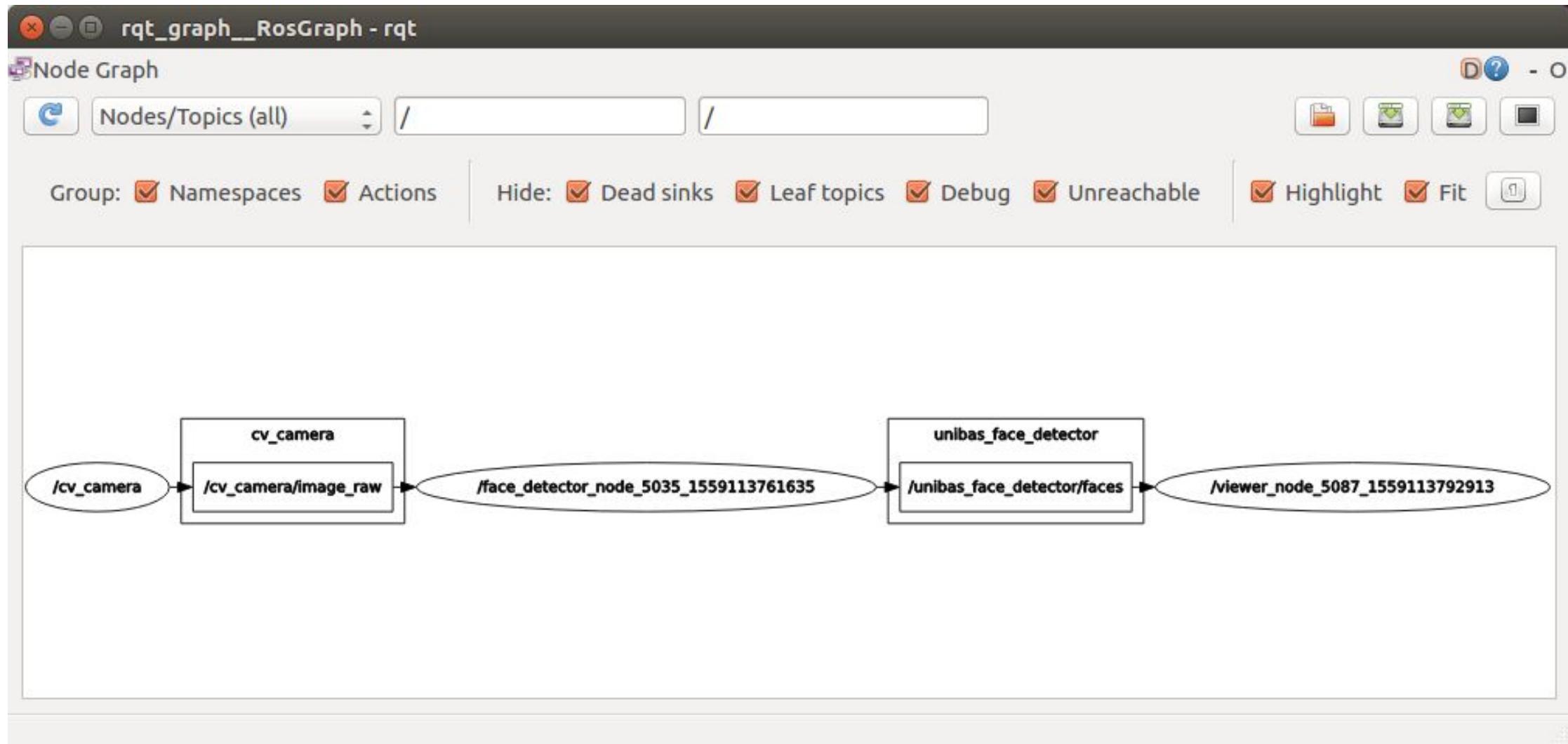
```
bloisi@bloisi-U36SG: ~
bloisi@bloisi-U36SG:~$ rosrun unibas_face_detector face_detector_node.py
[ INFO:0] Initialize OpenCL runtime...
```

# lanciamo il nodo viewer\_node

---



# rqt\_graph



# Esercizio 1

---

Utilizzare la rosbag people.bag

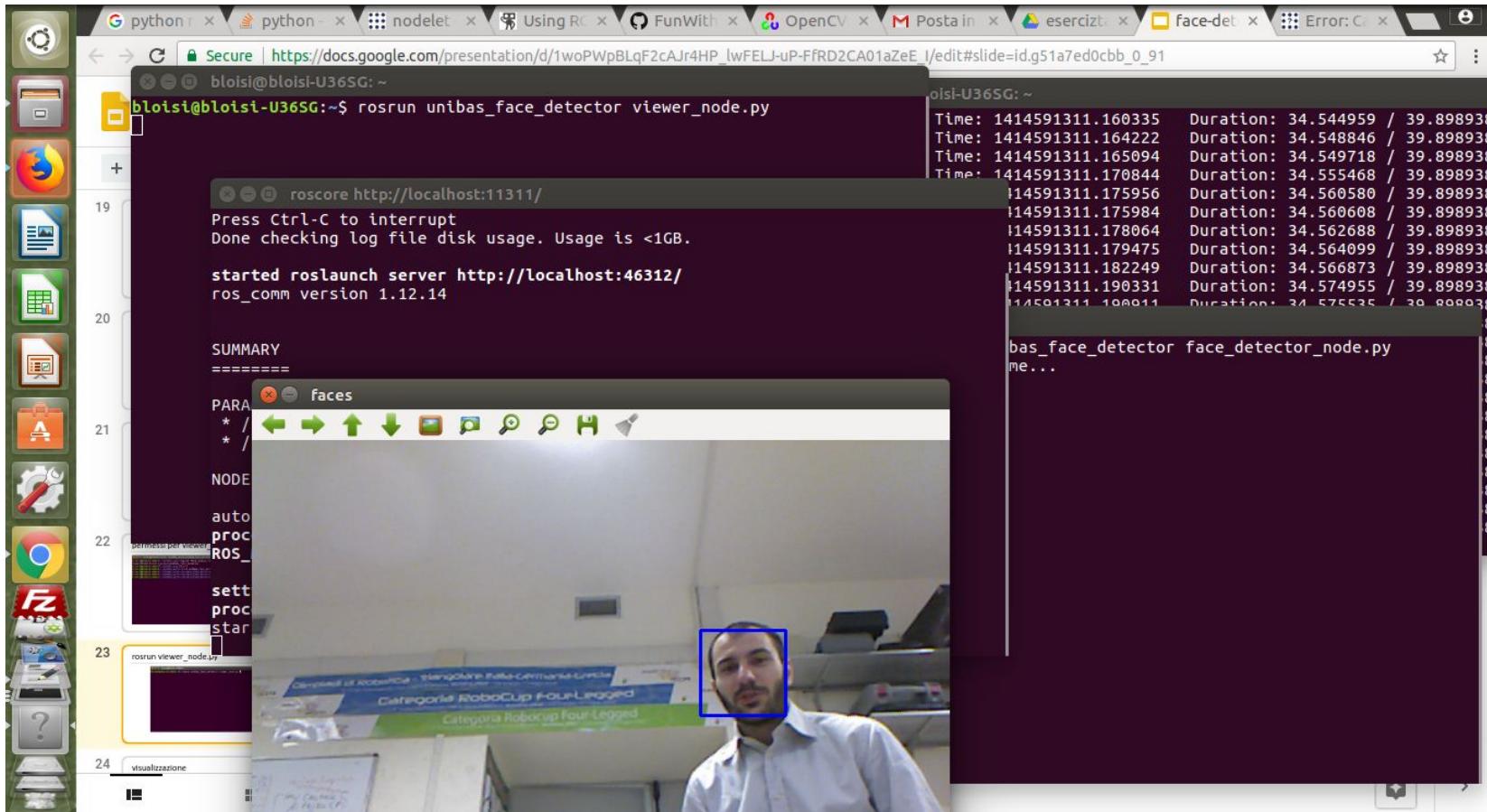
<https://drive.google.com/file/d/1oOMahlPdlwJkHMqXLtrLMktfx68-AGfJ/view?usp=sharing>

con il package unibas\_face\_detector

La detection dei volti presenti nella scena è corretta?

# Esercizio 2

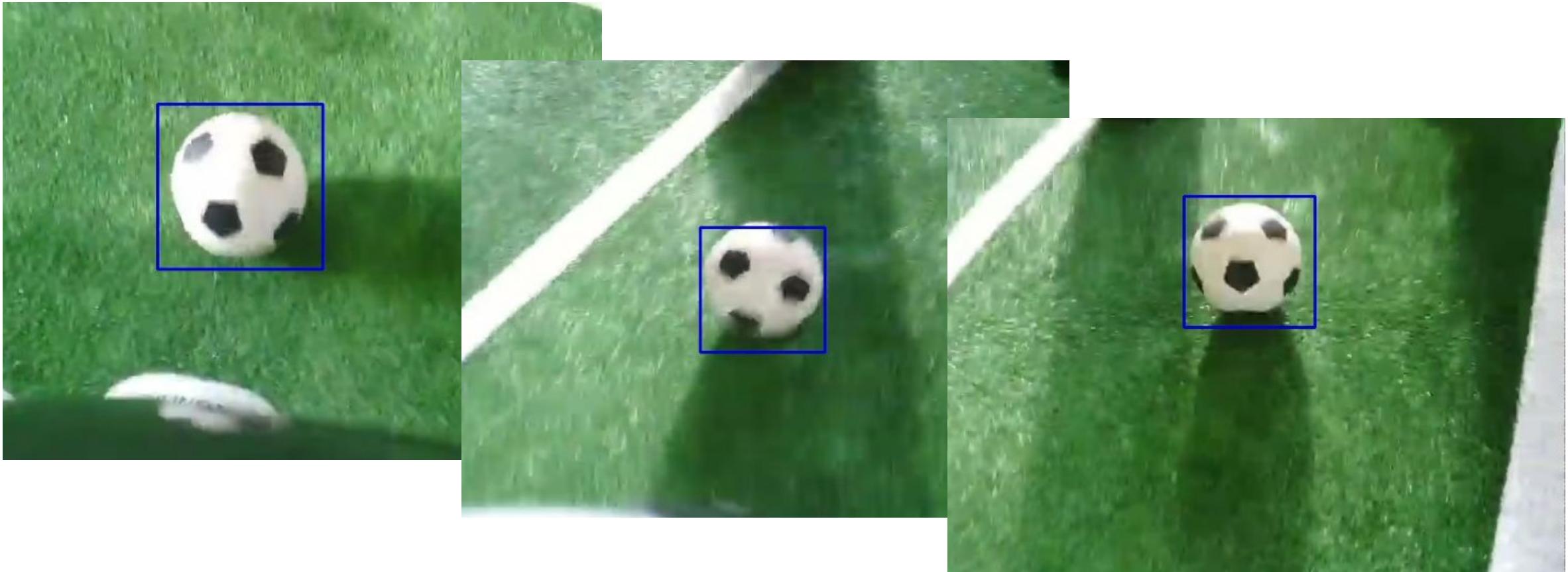
Creare un ros launchfile per evitare di dover aprire quattro differenti terminal per utilizzare il package unibas\_face\_detector



# Esercizio 3

---

Provare a utilizzare il codice di unibas\_face\_detector per creare un ball detector per i robot NAO che giocano a calcio



# Esercizio 3

---

Utilizzare

1. il file ball\_cascade.xml disponibile nel repository

<https://github.com/dbloisi/detectball>

2. la bag di test scaricabile al seguente indirizzo

<https://drive.google.com/file/d/1byC9QCq2ZomczZE-UgMedyNUqGtwI463/view?usp=sharing>

\* la bag di test è stata creata a partire dal video YouTube

<https://youtu.be/T1GUSiU-iro>



**UNIVERSITÀ DEGLI STUDI  
DELLA BASILICATA**

*Corso di Visione e Percezione*  
A.A. 2019/2020

Docente  
**Domenico Daniele Bloisi**

# Face detection in ROS

*Giugno 2020*

