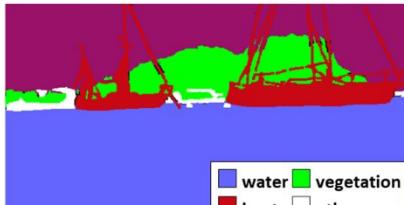
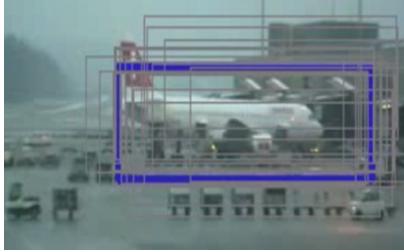
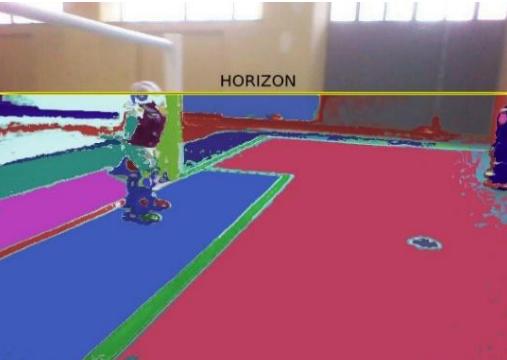
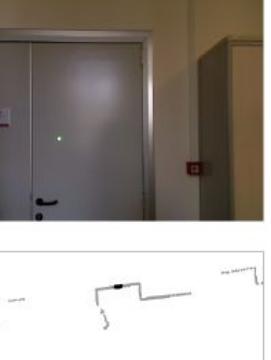




**UNIVERSITÀ DEGLI STUDI  
DELLA BASILICATA**

*Corso di Visione e Percezione*

# Esempi detection



# Domenico Daniele Bloisi

---

- Ricercatore RTD B

Dipartimento di Matematica, Informatica  
ed Economia

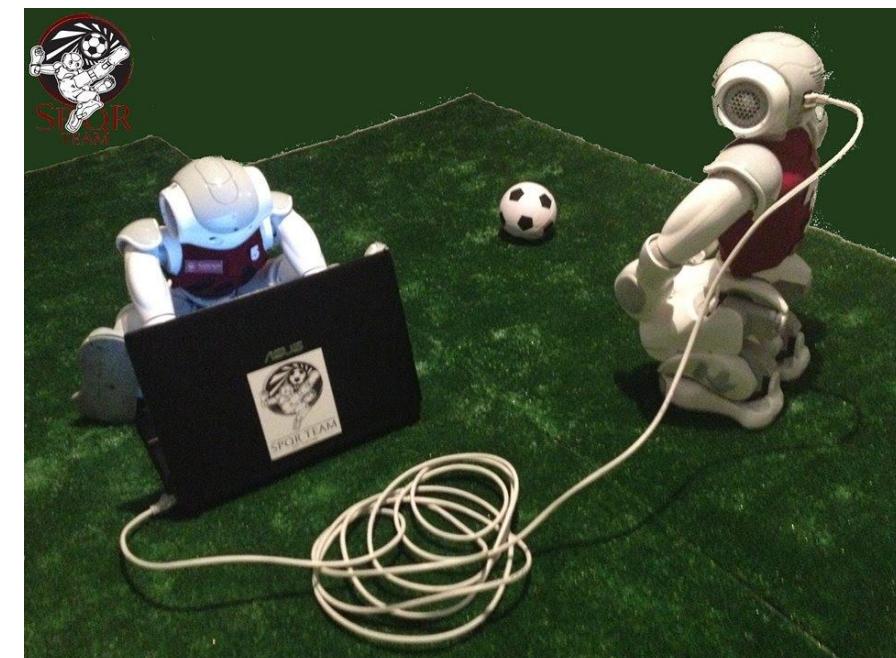
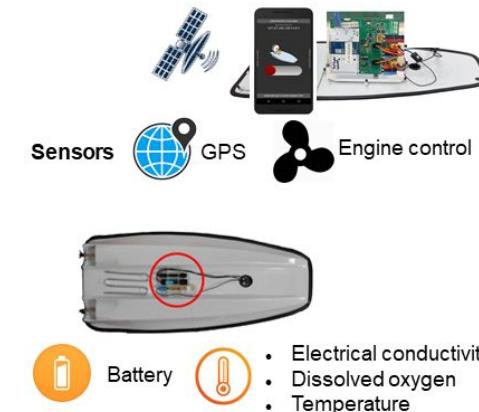
Università degli studi della Basilicata

<http://web.unibas.it/bloisi>

- SPQR Robot Soccer Team

Dipartimento di Informatica, Automatica  
e Gestionale Università degli studi di  
Roma “La Sapienza”

<http://spqr.diag.uniroma1.it>



# Informazioni sul corso

---

- Home page del corso  
<http://web.unibas.it/bloisi/corsi/visione-e-percezione.html>
- Docente: Domenico Daniele Bloisi
- Periodo: **Il semestre** marzo 2021 – giugno 2021

Martedì 17:00-19:00 (Aula COPERNICO)

Mercoledì 8:30-10:30 (Aula COPERNICO)



Codice corso Google Classroom:  
<https://classroom.google.com/c/Njl2MjA4MzgzNDFa?cjc=xgolays>

# Ricevimento

---

- Su appuntamento tramite Google Meet

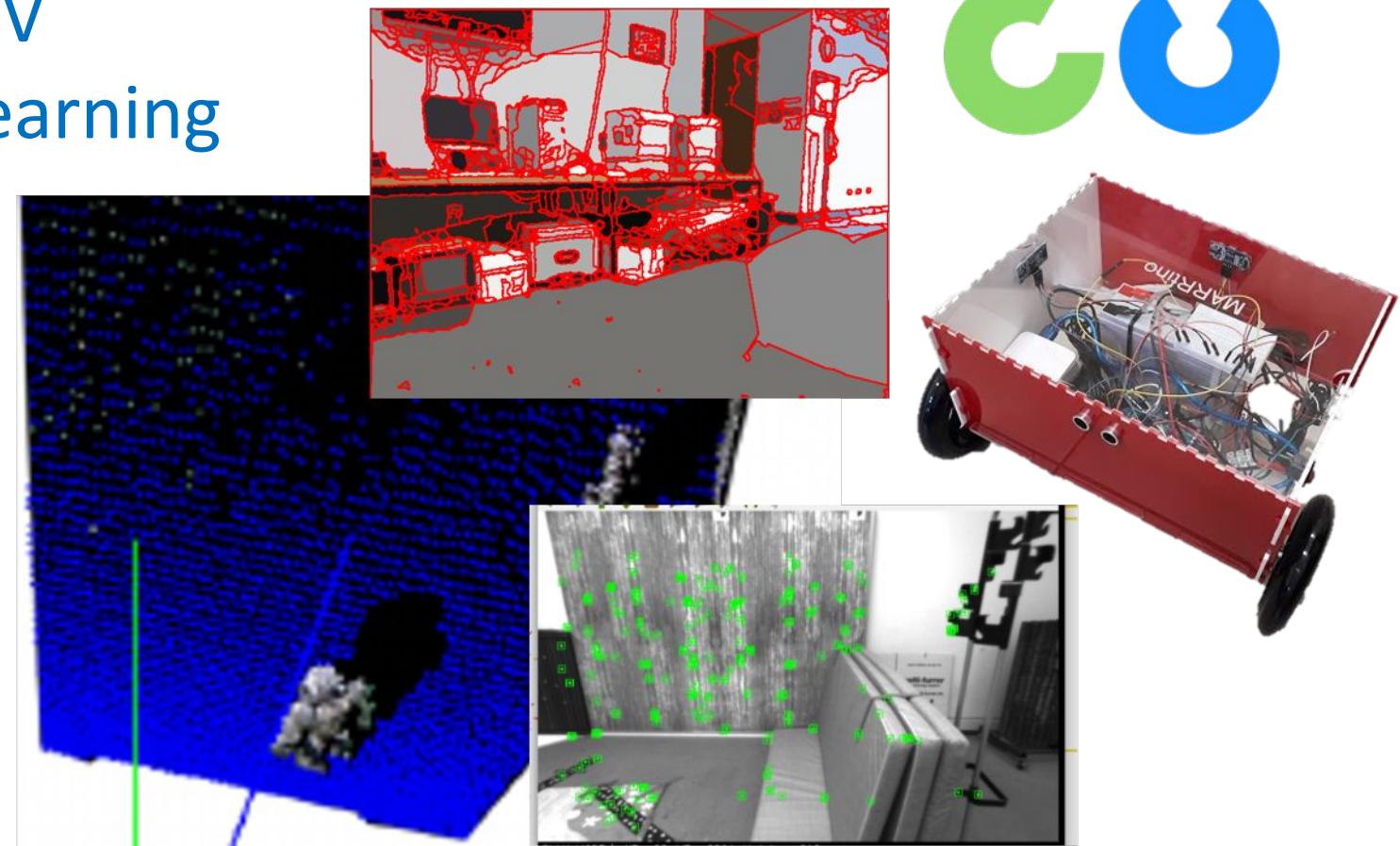
Per prenotare un appuntamento inviare  
una email a  
[domenico.bloisi@unibas.it](mailto:domenico.bloisi@unibas.it)



# Programma – Visione e Percezione

---

- Introduzione al linguaggio Python
- Elaborazione delle immagini con Python
- Percezione 2D – OpenCV
- Introduzione al Deep Learning
- ROS
- Il paradigma publisher and subscriber
- Simulatori
- Percezione 3D - PCL



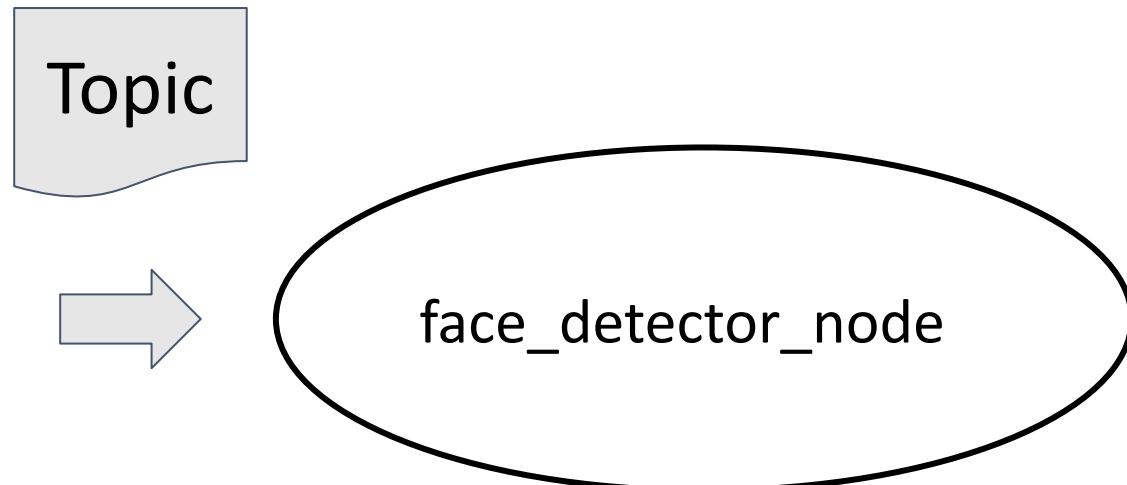
# Esempio face detection in ROS

---

- 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

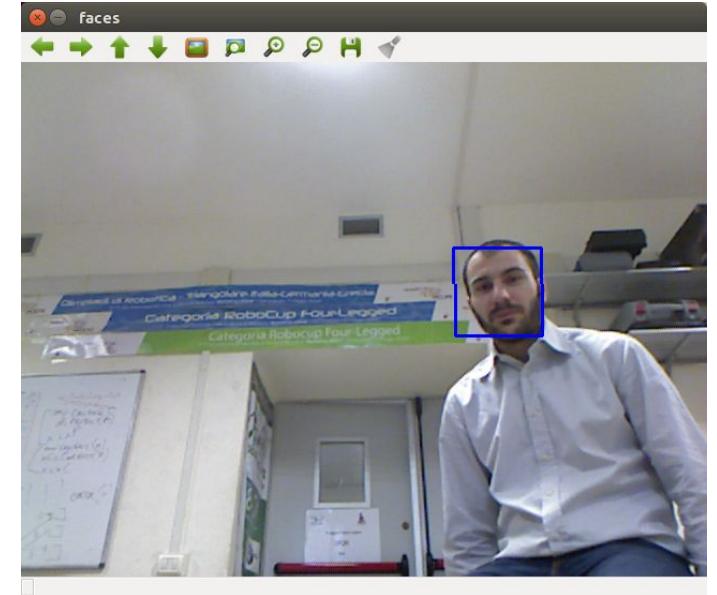
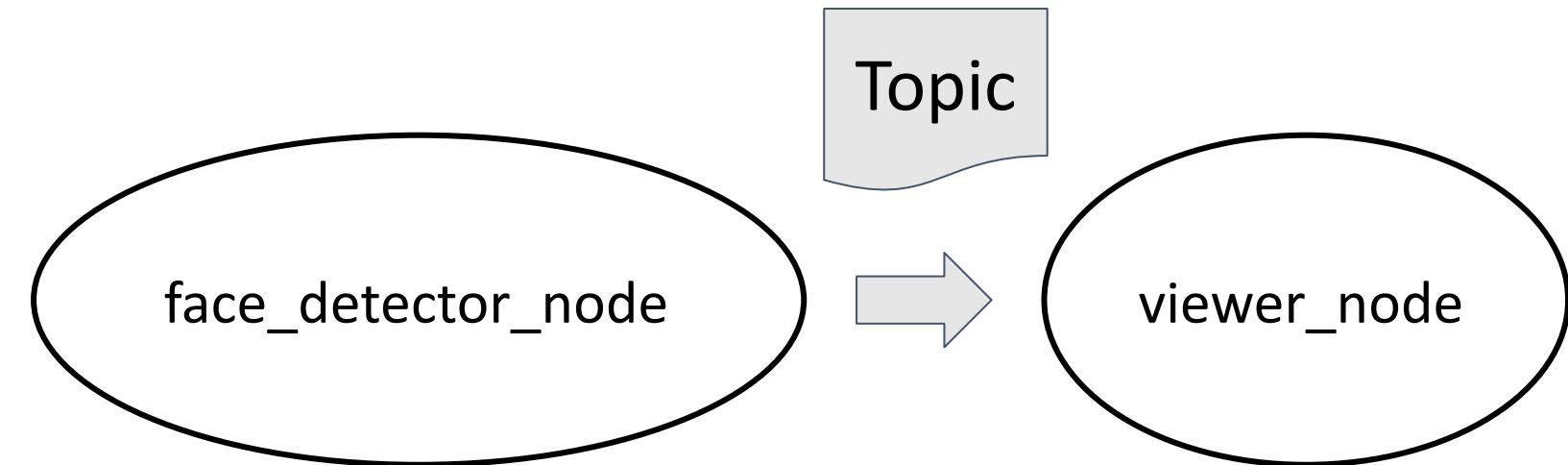
---



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

# face\_detector\_node e viewer\_node

---



visualizzazione  
immagine OpenCV

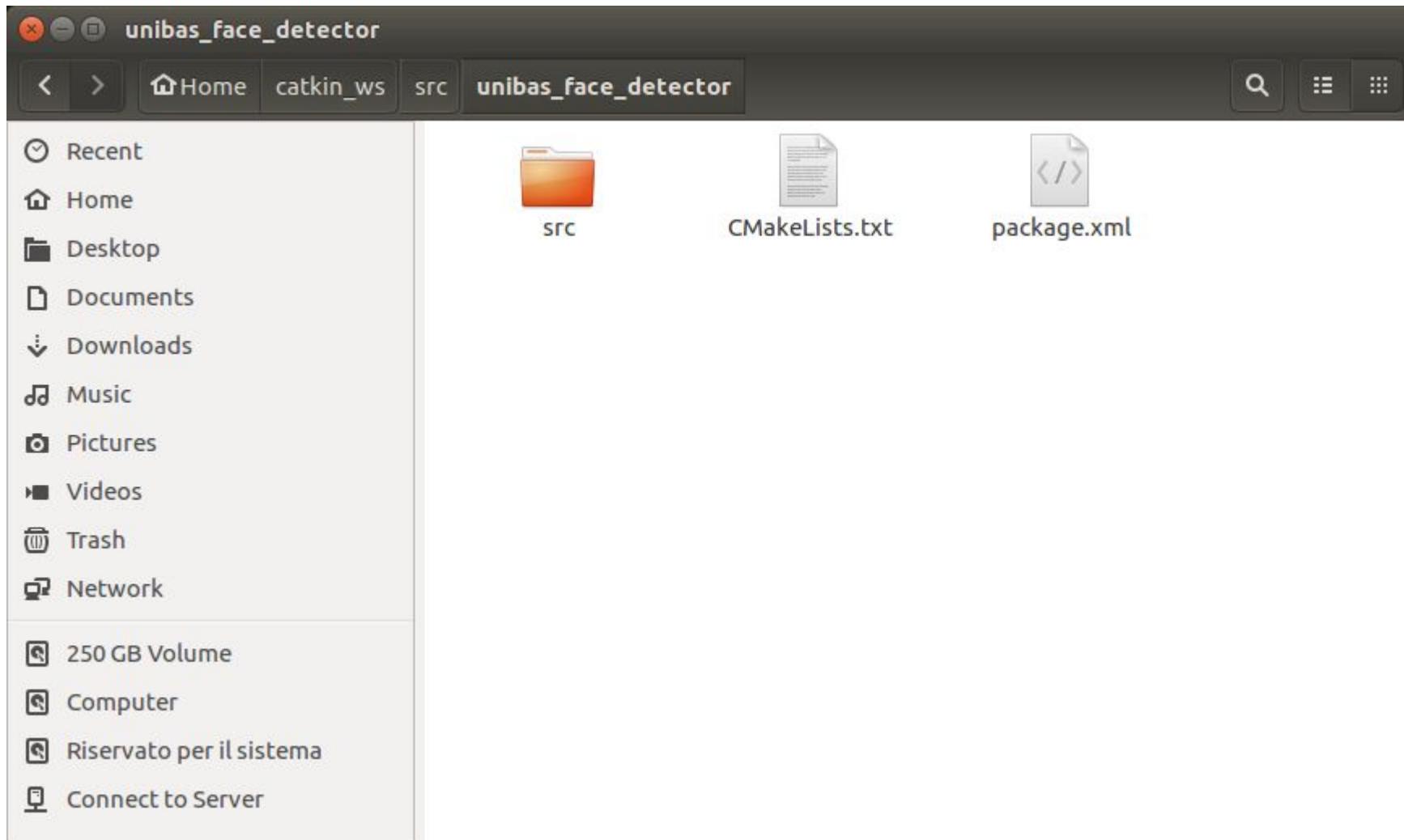
# creazione unibas\_face\_detector

---

```
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

---

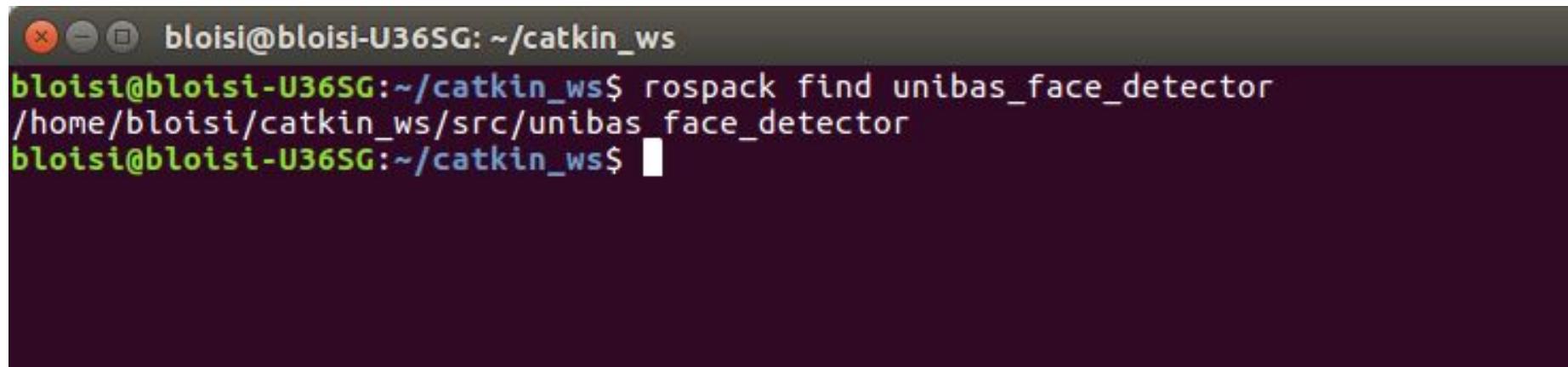
```
bloisi@bloisi-U36SG:~/catkin_ws
bloisi@bloisi-U36SG:~/catkin_ws/src$ catkin_create_pkg unibas_face_detector rospy
std_msgs sensor_msgs cv_bridge
Created file unibas_face_detector/package.xml
Created file unibas_face_detector/CMakeLists.txt
Created folder unibas_face_detector/src
Successfully created 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=/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

```
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

---



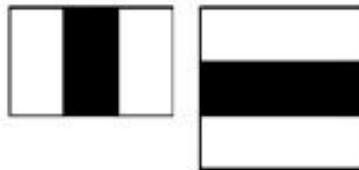
```
bloisi@bloisi-U36SG: ~/catkin_ws
bloisi@bloisi-U36SG:~/catkin_ws$ rospack find unibas_face_detector
/home/bloisi/catkin_ws/src/unibas_face_detector
bloisi@bloisi-U36SG:~/catkin_ws$ █
```

rospack find unibas face detector

# ci serve un detector di volti



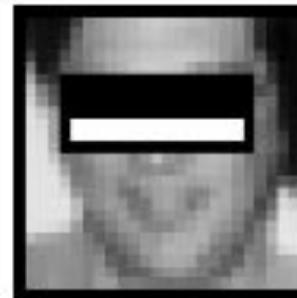
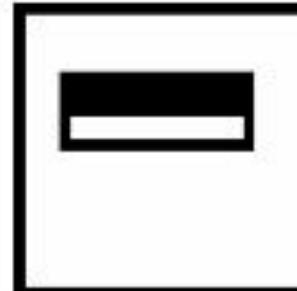
(a) Edge Features



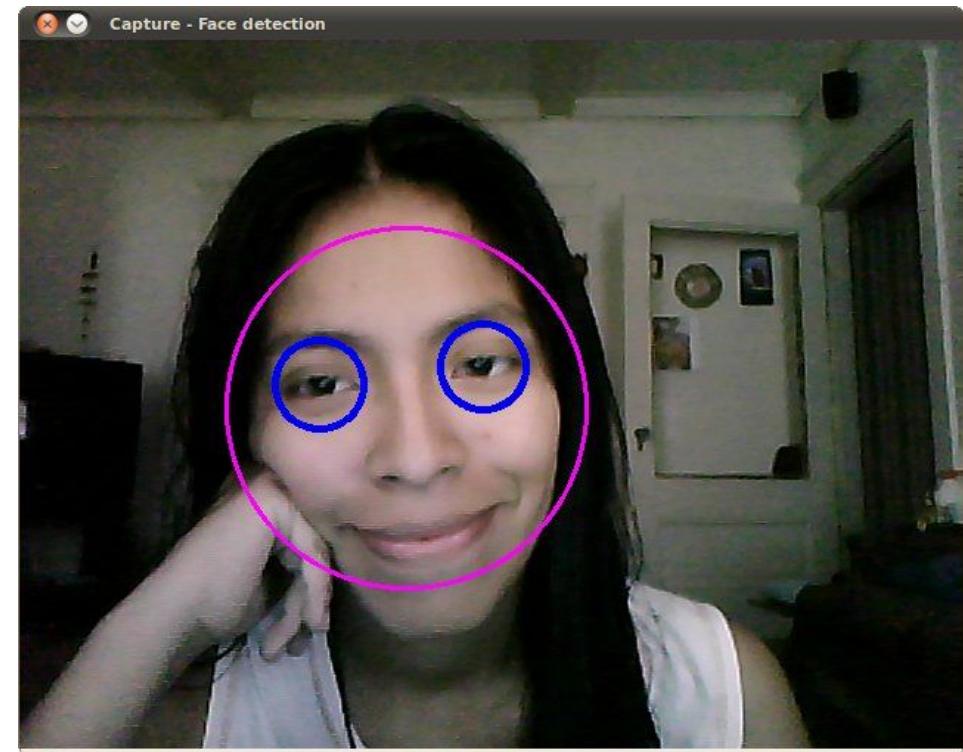
(b) Line Features



(c) Four-rectangle features



Face detection and eye detection  
using the Haar Feature-based  
Cascade Classifiers



# useremo un modello già addestrato

3.4 · opencv / data / haarcascades / Go to file

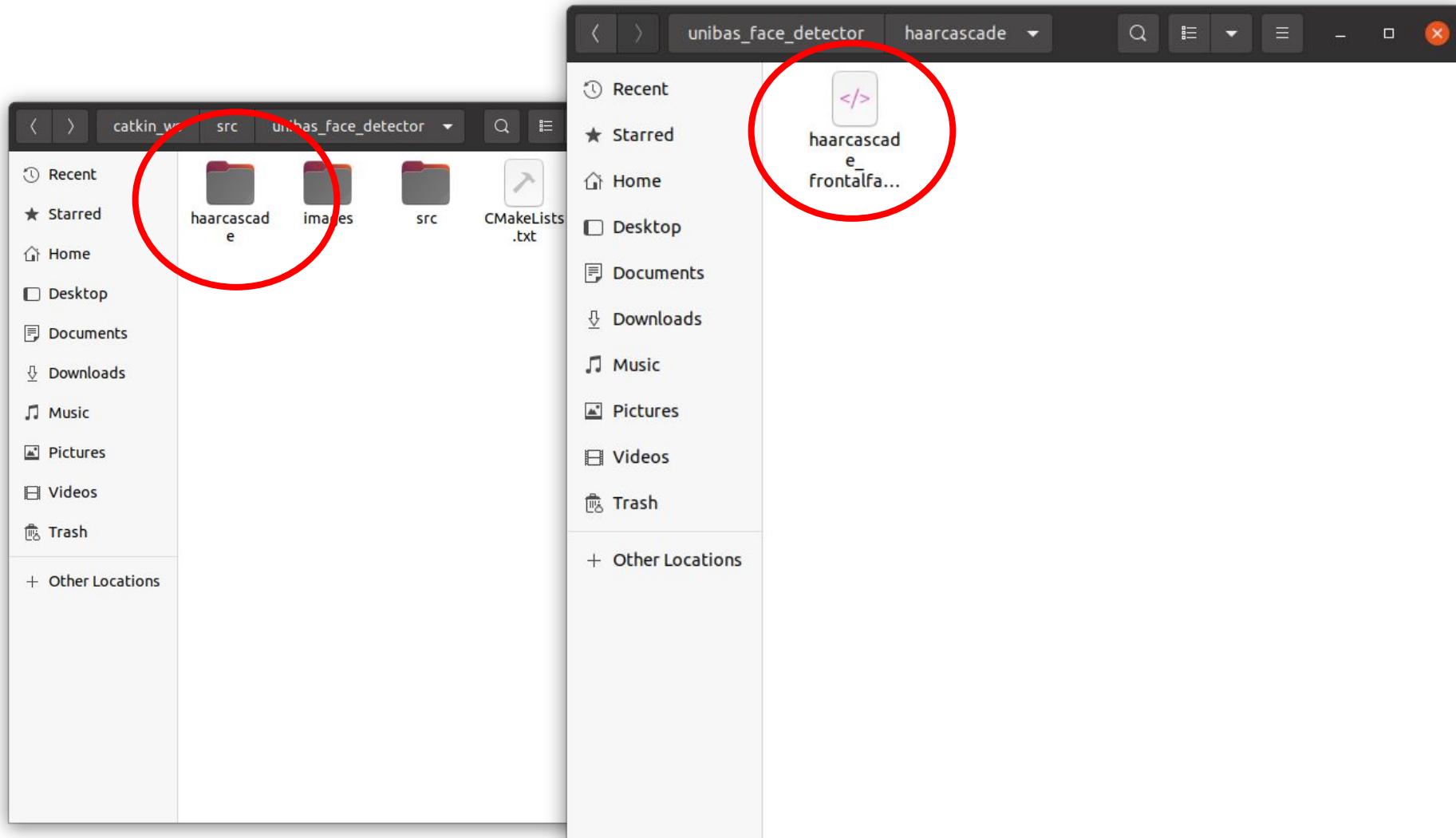
This branch is 2227 commits behind master. Pull request · Compare

alalek fix files permissions ✓ reffesa · on Apr 13, 2020 · History

File	Description	Time Ago
haarcascade_eye.xml	some attempts to tune the performance	8 years ago
haarcascade_eye_tree_eyeglasses.xml	some attempts to tune the performance	8 years ago
haarcascade_frontalcatface.xml	fix files permissions	14 months ago
haarcascade_frontalcatface_extended.xml	fix files permissions	14 months ago
haarcascade_frontalface_alt.xml	some attempts to tune the performance	8 years ago
haarcascade_frontalface_alt2.xml	some attempts to tune the performance	8 years ago
haarcascade_frontalface_alt_tree.xml	some attempts to tune the performance	8 years ago
haarcascade_frontalface_default.xml	some attempts to tune the performance	8 years ago
haarcascade_fullbody.xml	Some mist. typo fixes	3 years ago
haarcascade_lefteye_2splits.xml	some attempts to tune the performance	8 years ago
haarcascade_licence_plate_rus_16stages.xml	Added Haar cascade for russian cars licence plate detection, 16 stage...	7 years ago
haarcascade_lowerbody.xml	Some mist. typo fixes	3 years ago
haarcascade_profileface.xml	some attempts to tune the performance	8 years ago
haarcascade_righteye_2splits.xml	some attempts to tune the performance	8 years ago
haarcascade_russian_plate_number.xml	Create haarcascade_russian_plate_number.xml	7 years ago
haarcascade_smile.xml	fixing models to resolve XML violation issue	4 years ago
haarcascade_upperbody.xml	Some mist. typo fixes	3 years ago

# inseriamolo nel progetto

---



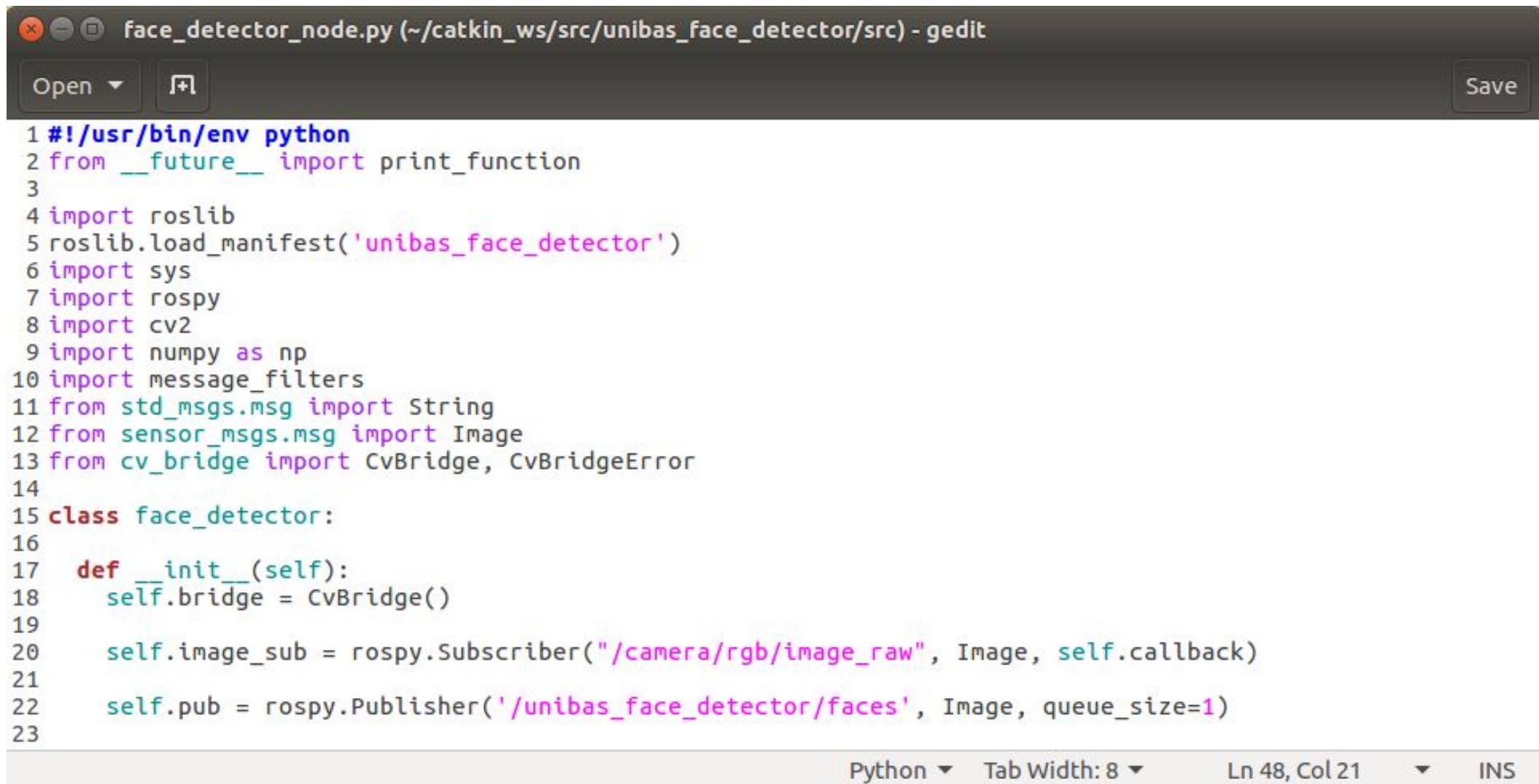
# creiamo face\_detector\_node.py

---



# codice face\_detector\_node.py

---



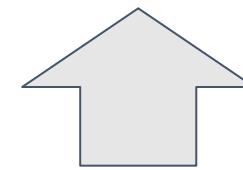
The screenshot shows a Gedit text editor window with the title "face\_detector\_node.py (~/catkin\_ws/src/unibas\_face\_detector/src) - gedit". The code is a Python script for a ROS node. The code uses ROS message types from std\_msgs, sensor\_msgs, and cv\_bridge. It defines a class "face\_detector" with an \_\_init\_\_ method that initializes a CvBridge and a subscriber to /camera/rgb/image\_raw. It also creates a publisher to /unibas\_face\_detector/faces.

```
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

```
Open ▾  face_detector_node.py ~ /catkin_ws/src/unibas_Face_detector/src Save ⌂ ×  
22     self.pub = rospy.Publisher('/unibas_face_detector/faces', Image, queue_size=1)  
23  
24 def callback(self, rgb_data):  
25  
26     try:  
27         img = self.bridge.imgmsg_to_cv2(rgb_data, "bgr8")  
28         face_cascade = cv2.CascadeClassifier('/home/bloisi/catkin_ws/src/unibas_Face_detector/haarcascade/-  
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
```



haarcascade filename

# 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 menu bar includes "Open" and "Save" buttons. 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 file is saved in "Python" mode, with a tab width of 8, and the current position is "Ln 48, Col 21". There is also 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

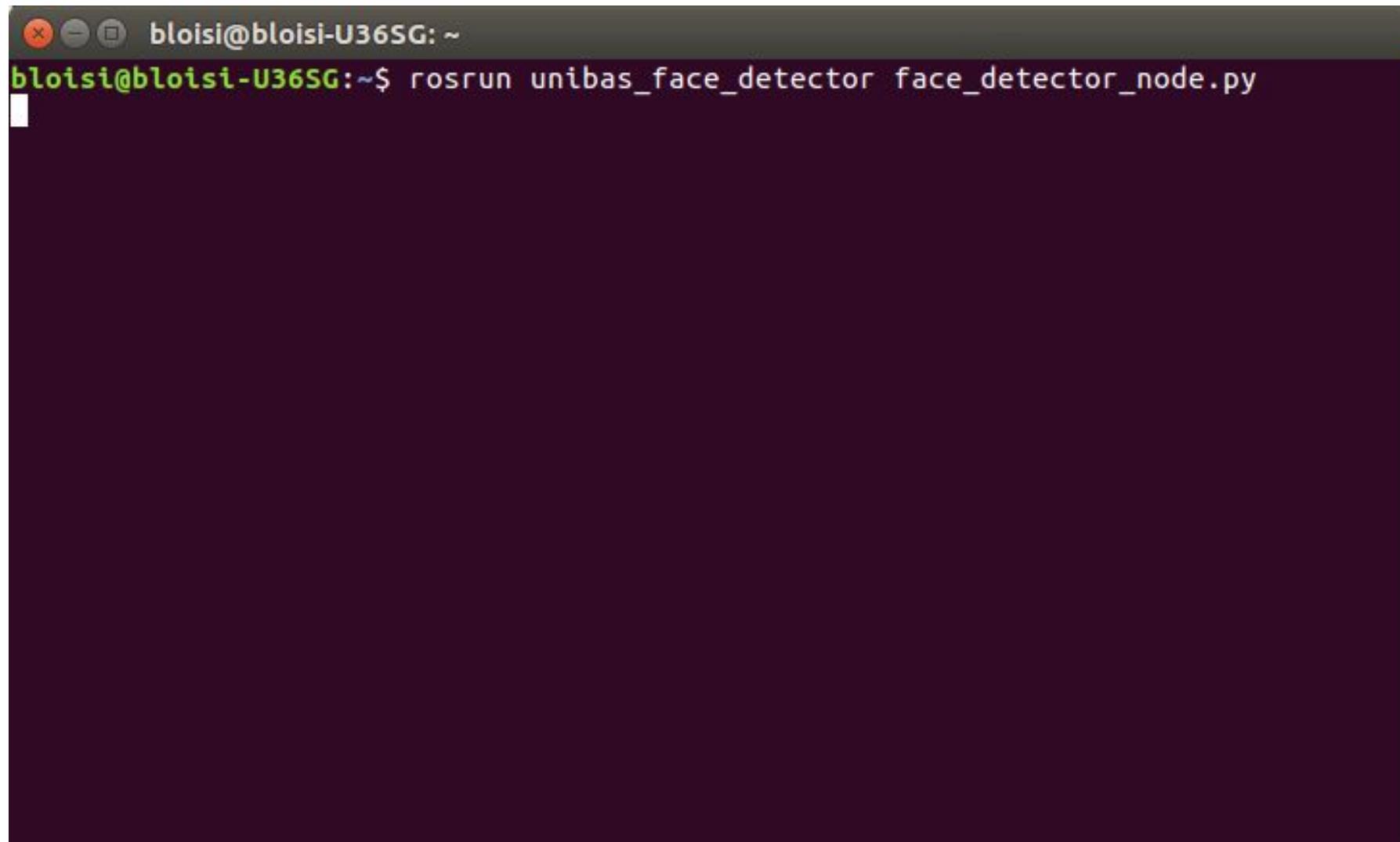
NODES

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

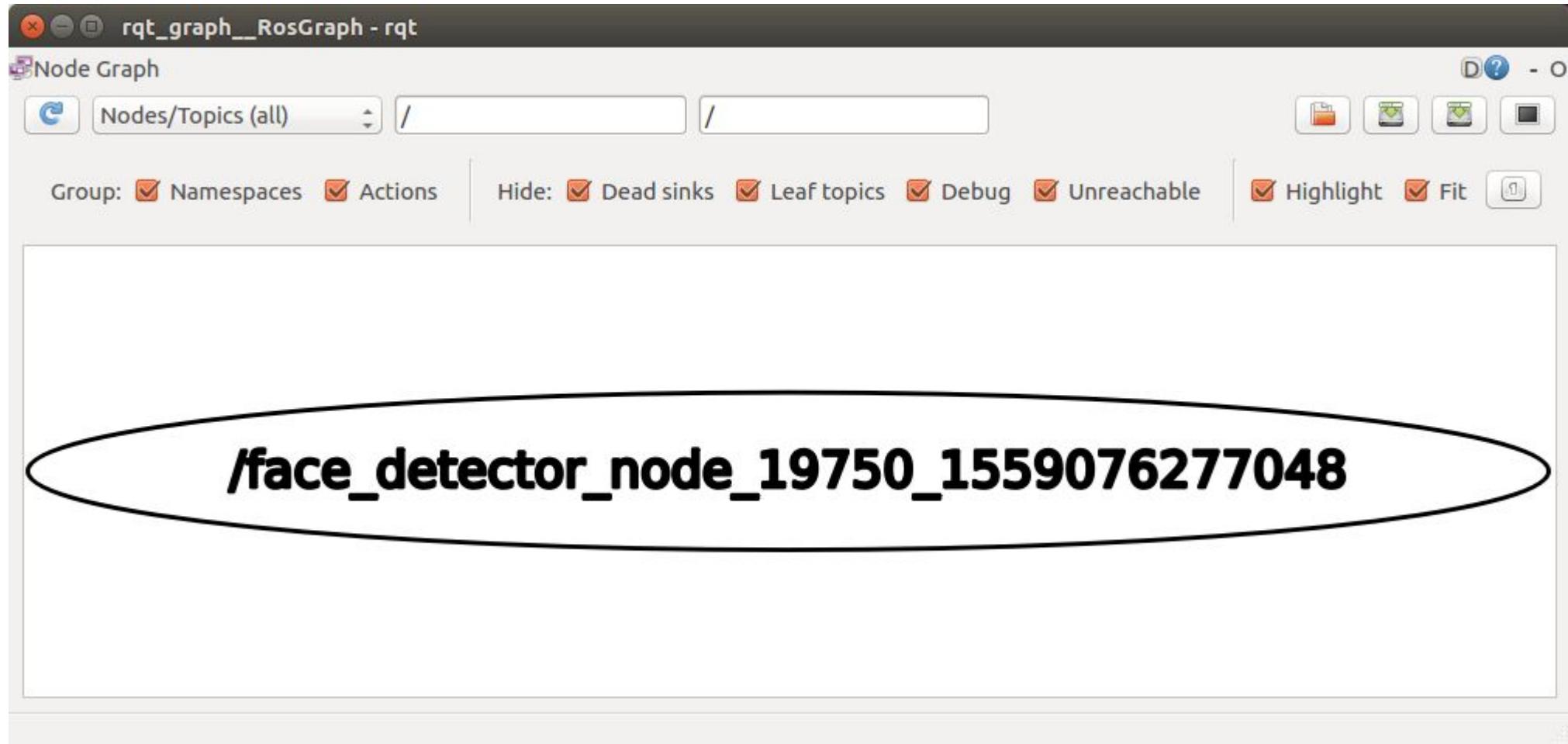
---

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

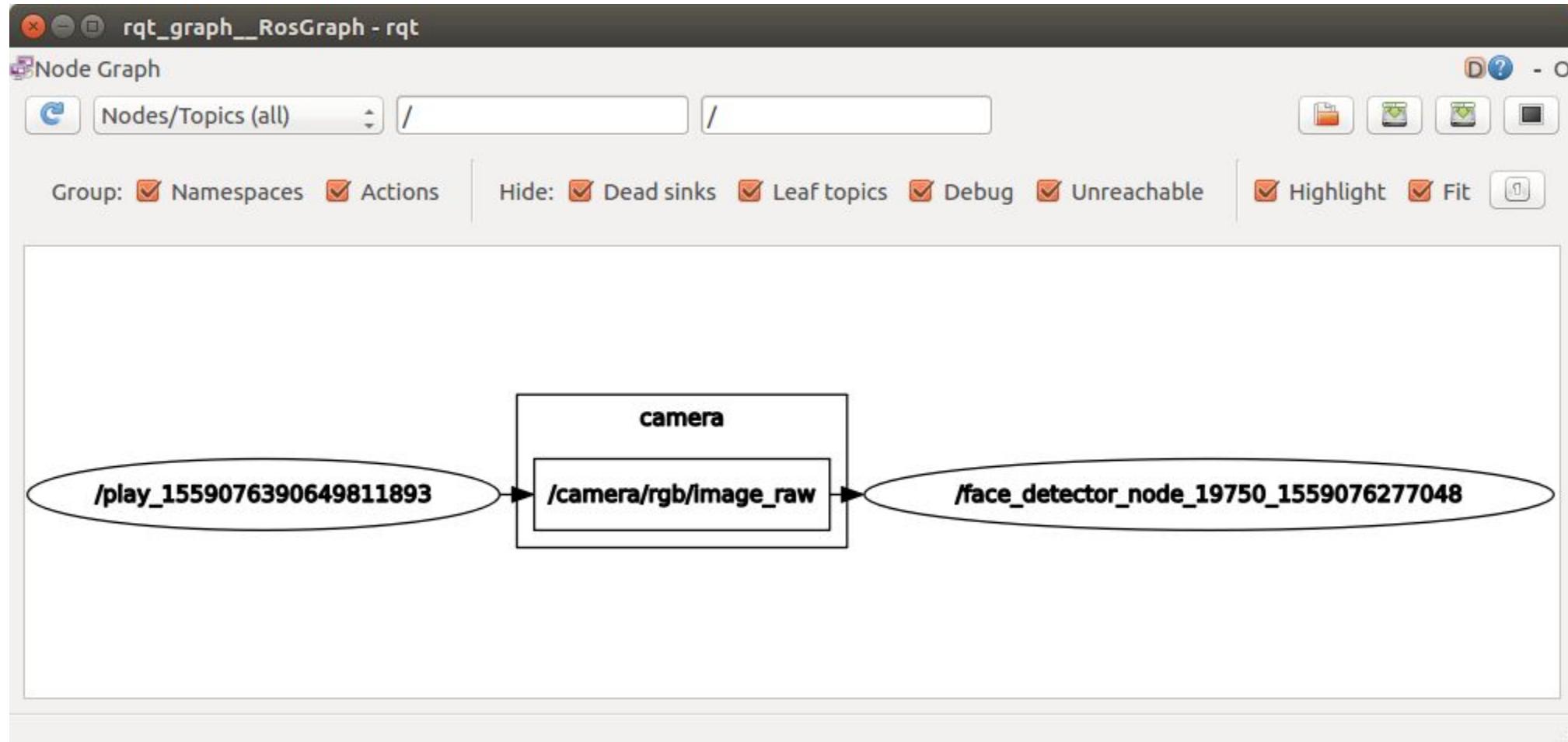
---

```
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.

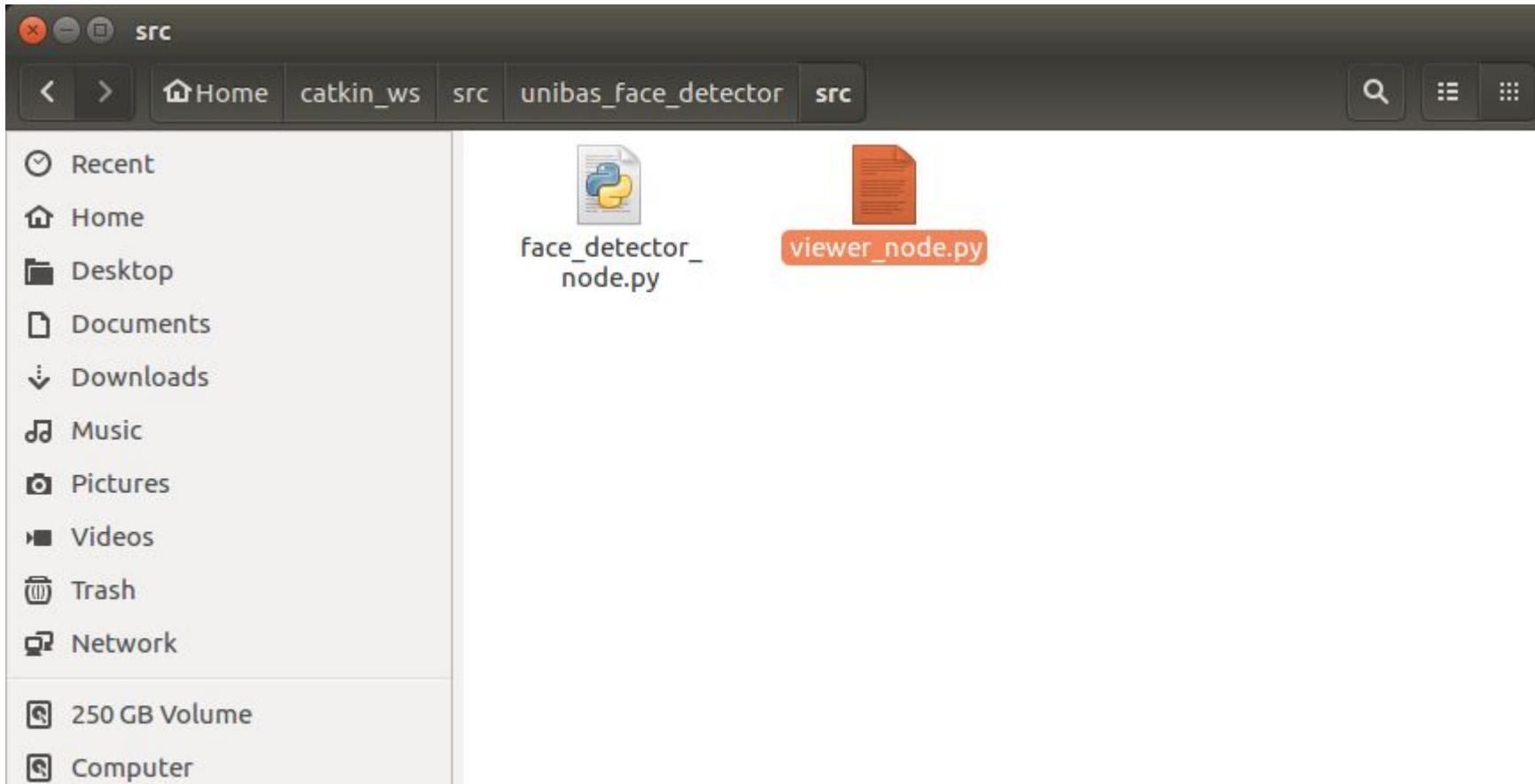
Hit space to toggle paused, or 's' to step.
[RUNNING] Bag Time: 1414591276.615376 Duration: 0.000000 / 39.898938
[RUNNING] Bag Time: 1414591276.784976 Duration: 0.169601 / 39.898938
[RUNNING] Bag Time: 1414591276.802653 Duration: 0.187277 / 39.898938
[RUNNING] Bag Time: 1414591276.804009 Duration: 0.188634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



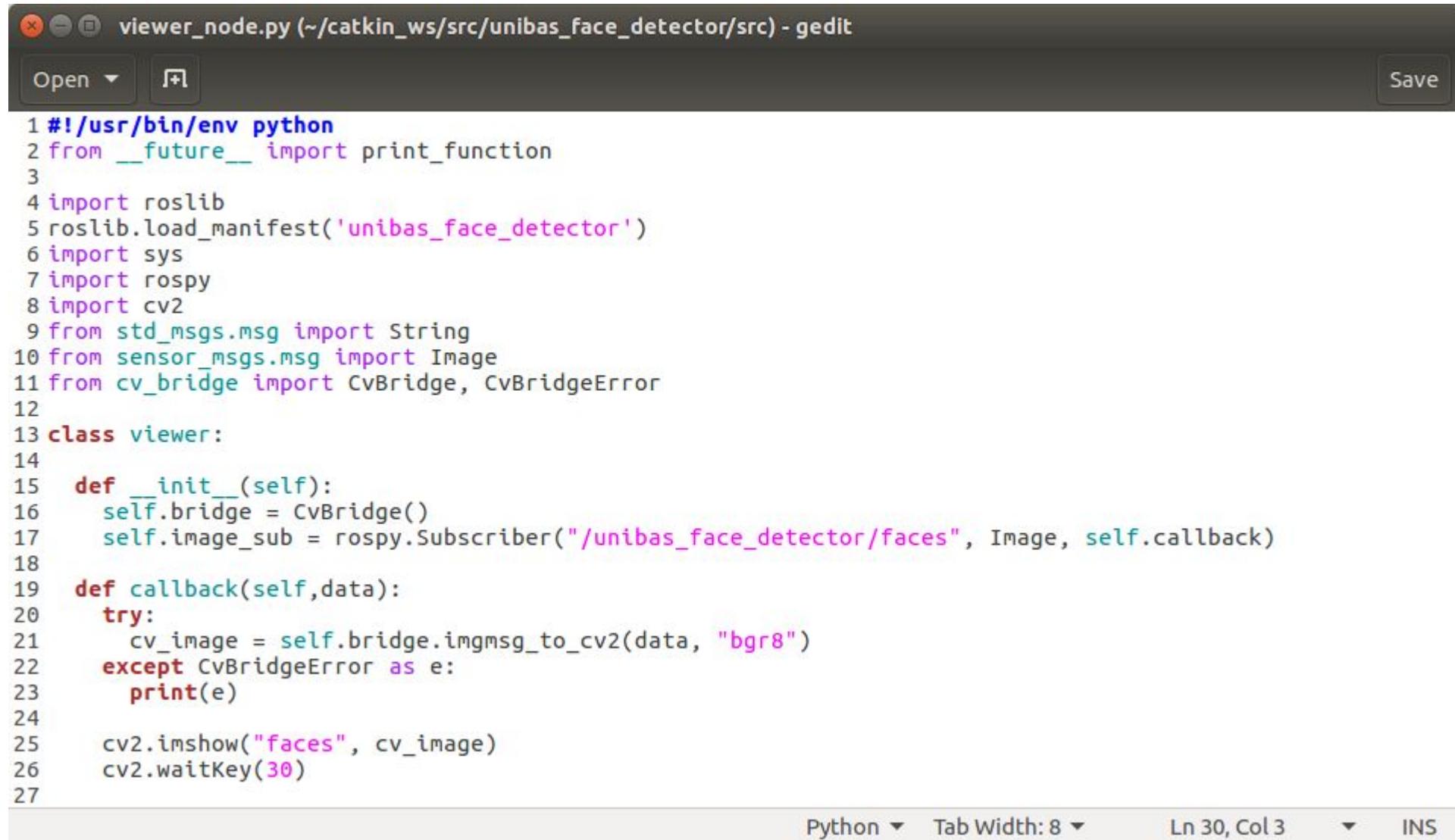
# viewer\_node

---



# codice viewer\_node

---



```
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

---



A screenshot of a terminal window titled "viewer\_node.py (~/catkin\_ws/src/unibas\_face\_detector/src) - gedit". The window contains Python code for a ROS node named "viewer\_node". The code defines a main function that initializes a ROS node, creates a viewer, and spins. It handles a KeyboardInterrupt exception by shutting down the node and destroying all windows. The code also includes a check for the \_\_name\_\_ variable to ensure it runs as the main module. The terminal shows the output of the node running, with messages indicating it is processing frames and has detected faces.

```
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

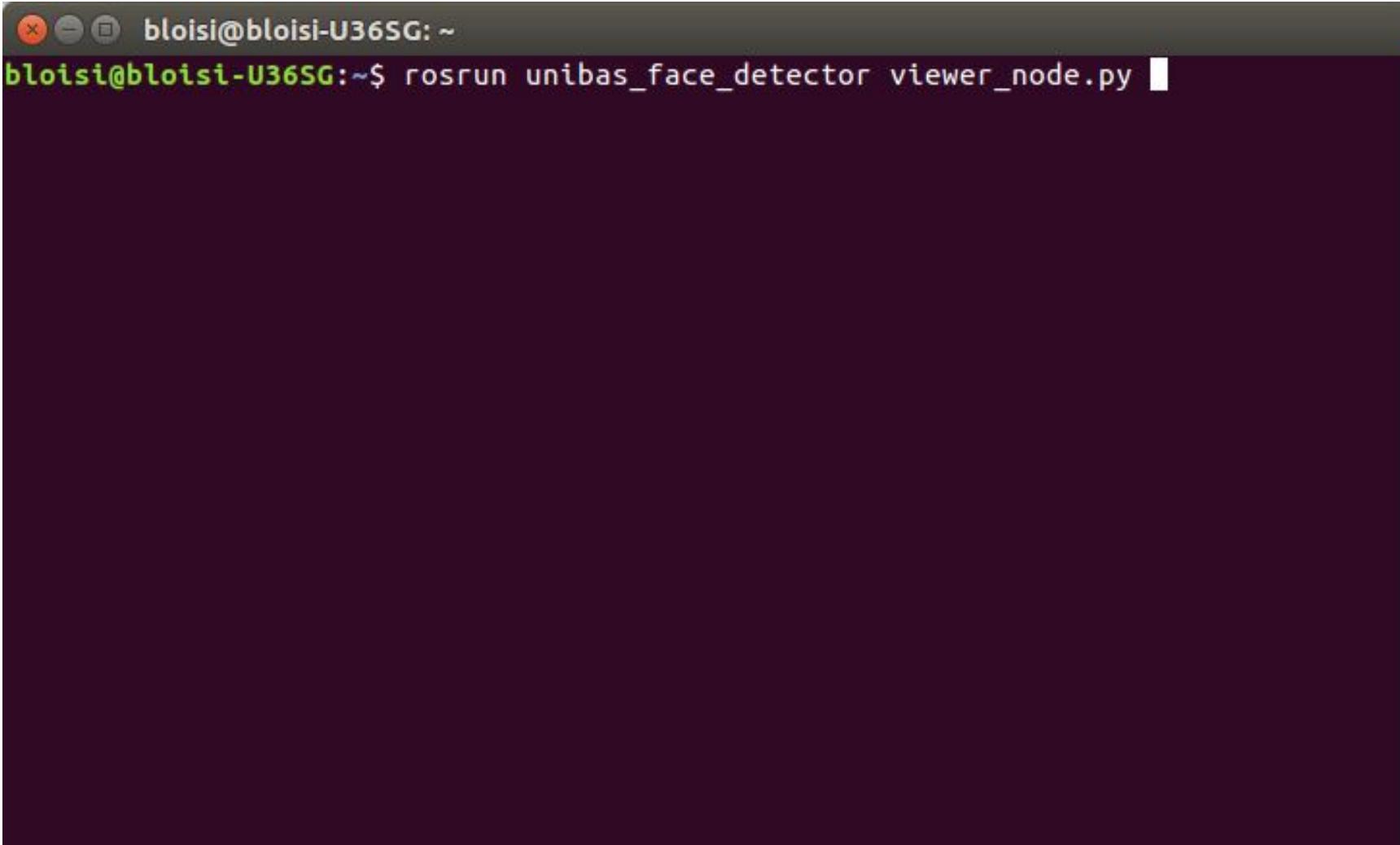
# permessi per viewer\_node.py

---

```
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$ █
```

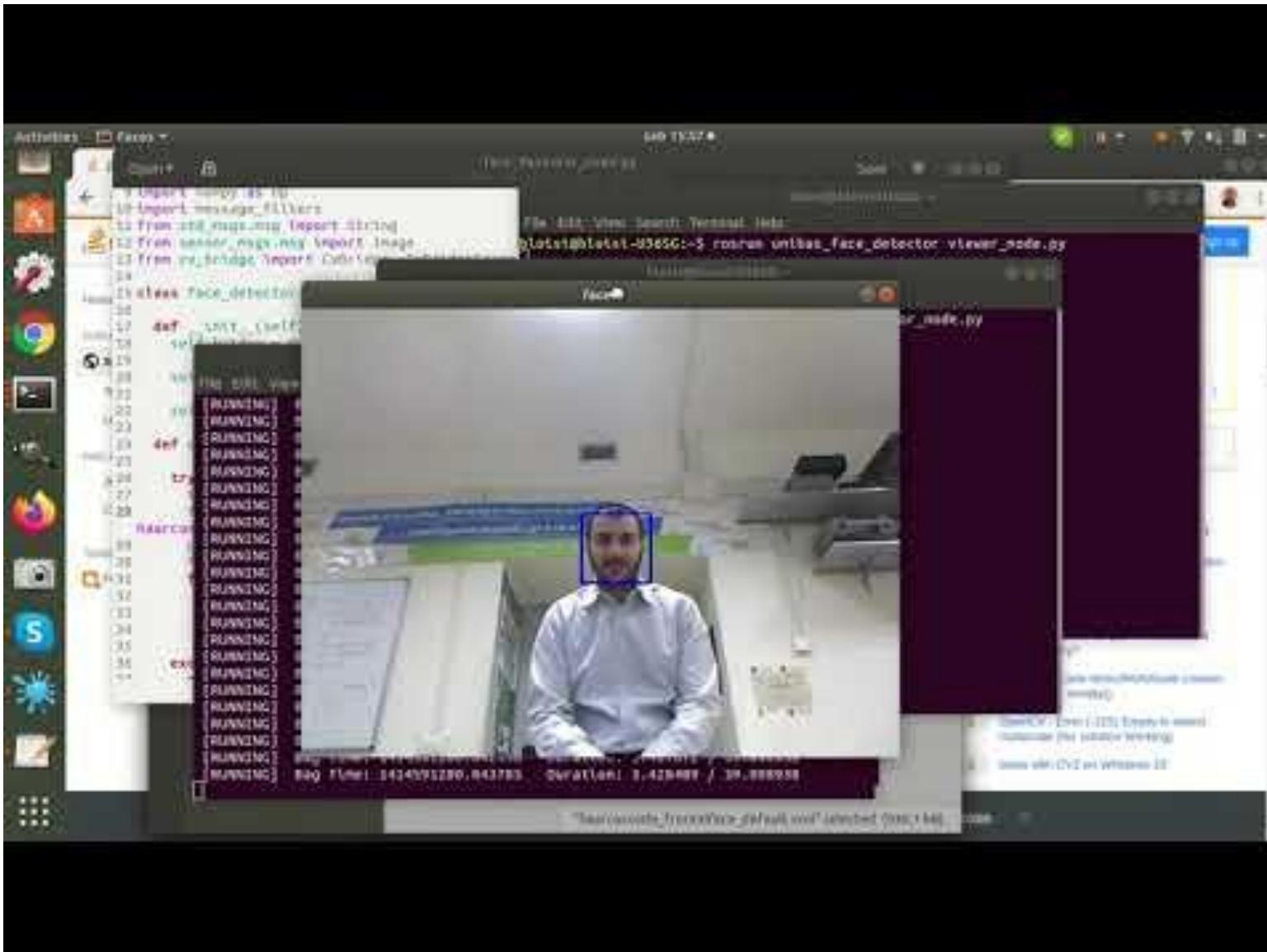
# rosrun viewer\_node.py

---

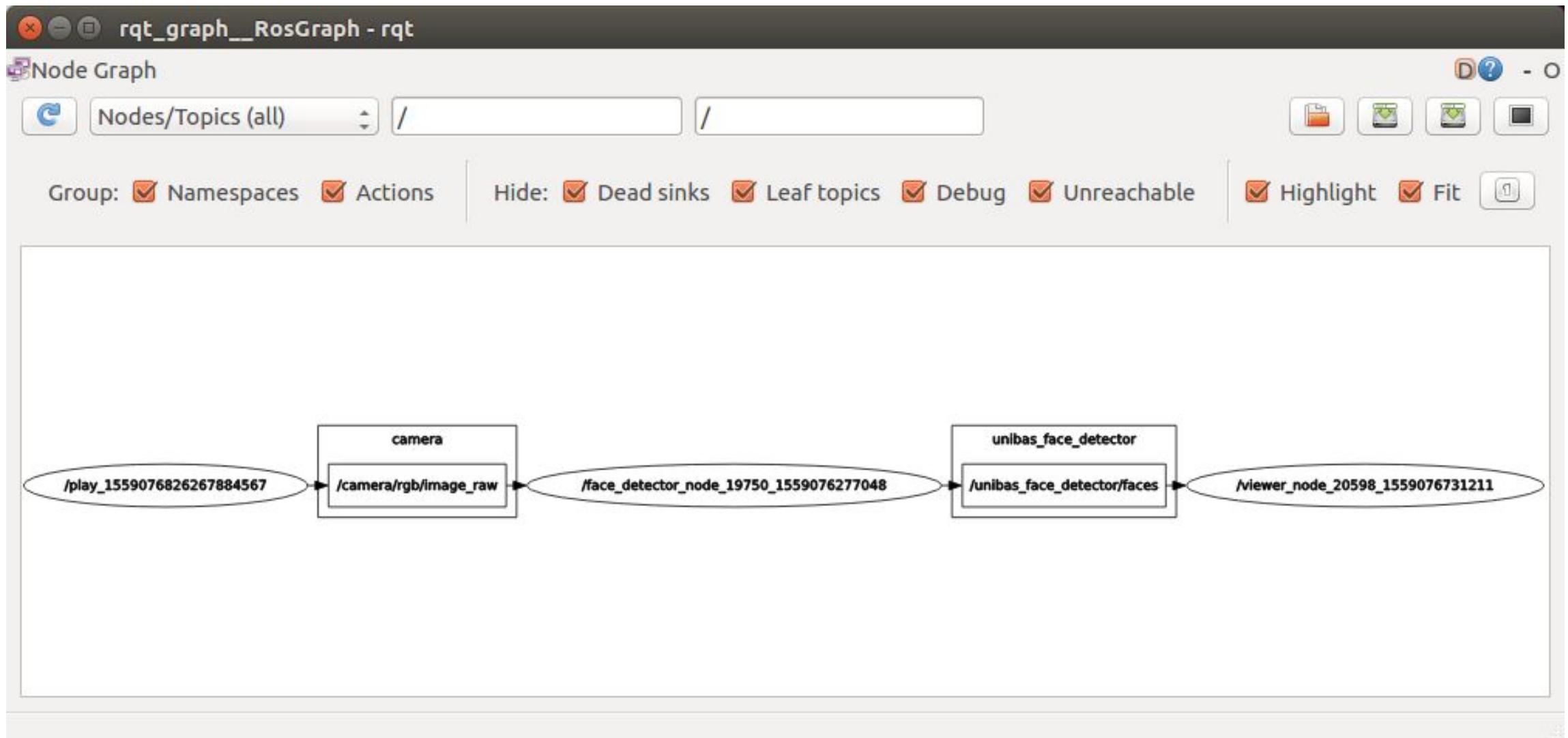
A screenshot of a terminal window titled "bloisi@bloisi-U36SG: ~". The window contains a single line of text: "bloisi@bloisi-U36SG:~\$ rosrun unibas\_face\_detector viewer\_node.py". The terminal has a dark background with light-colored text and standard window controls at the top.

# visualizzazione

---



# rqt\_graph

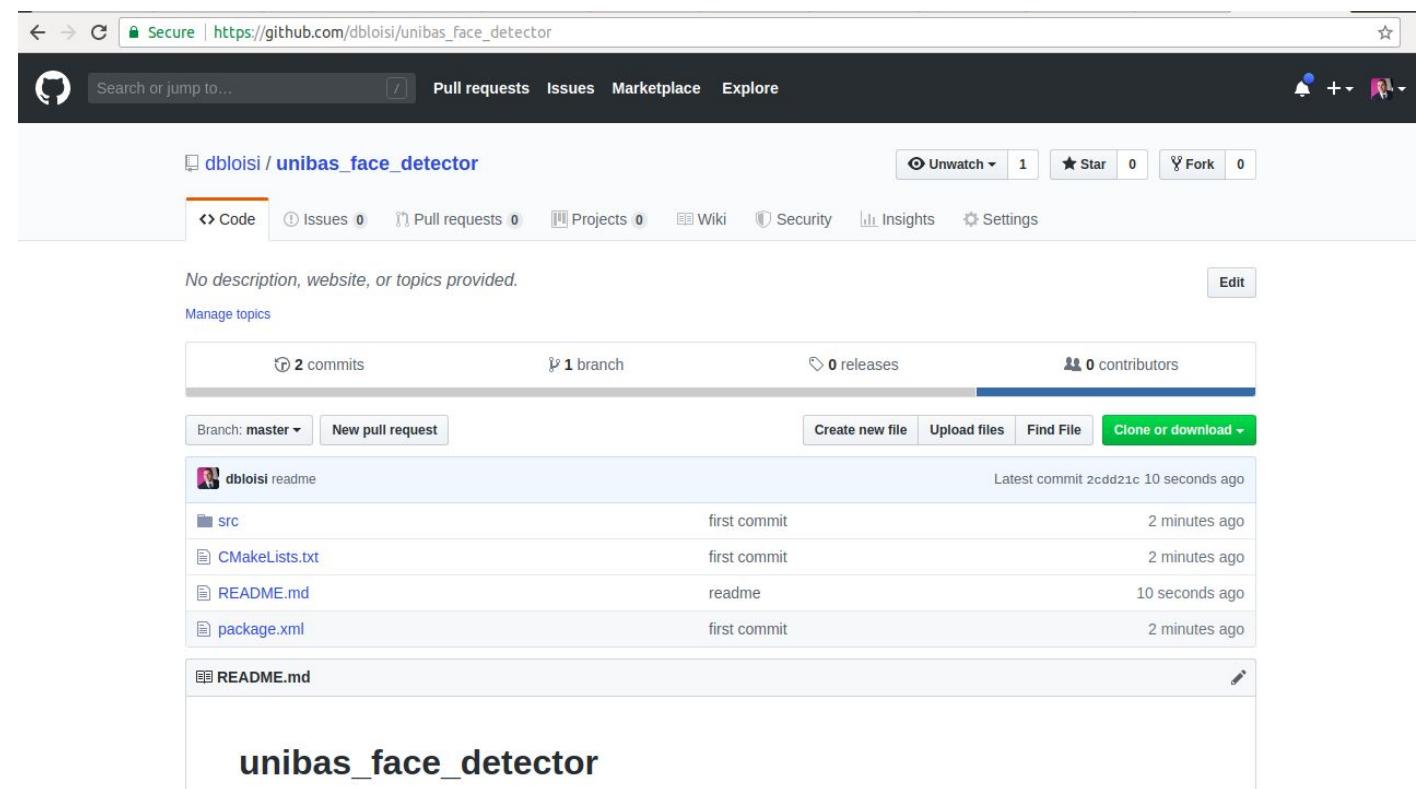


# repository unibas\_face\_detector

---

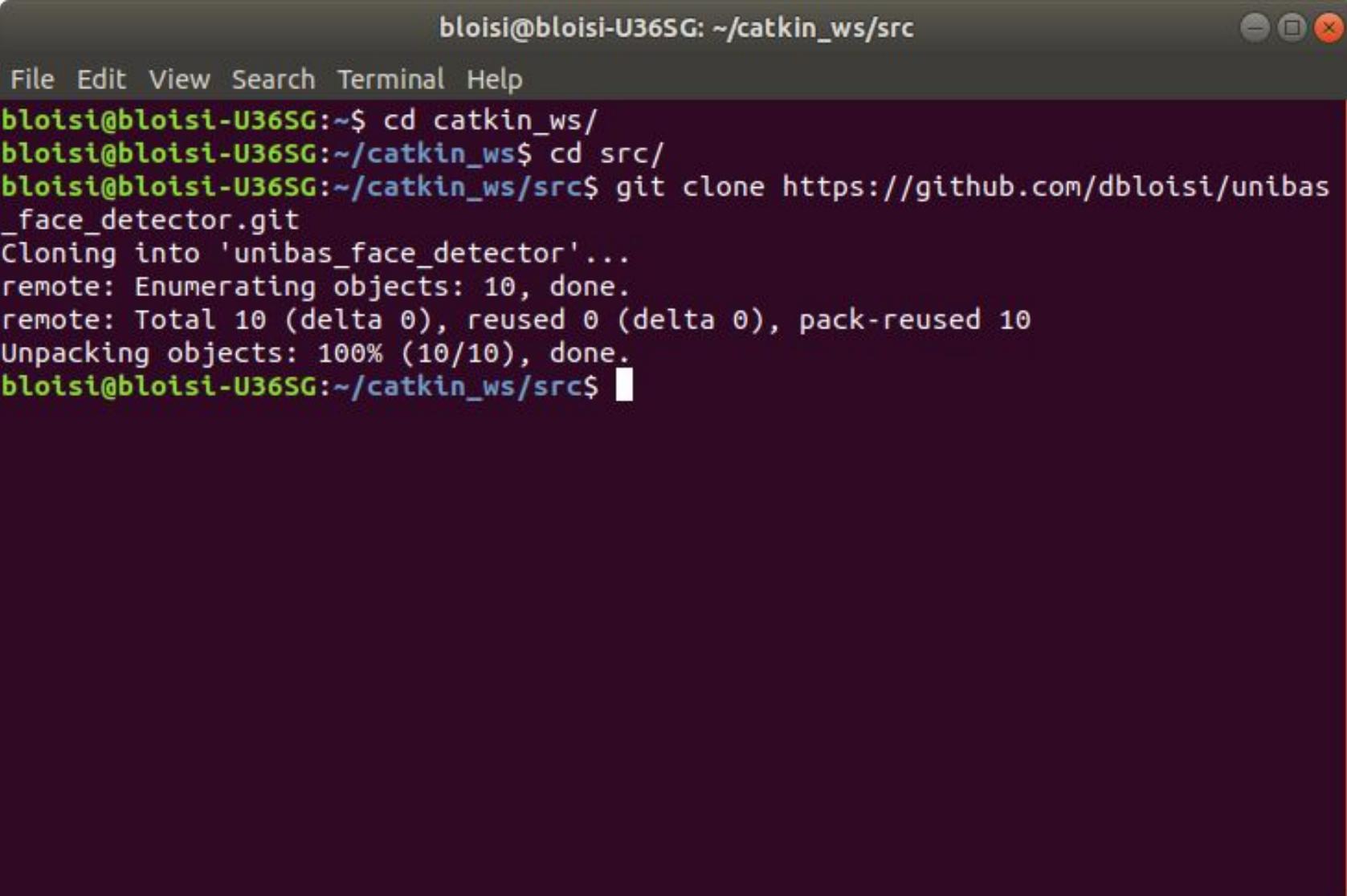
Il codice del repository `unibas_face_detector` è disponibile al seguente link

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



# cloniamo 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 shows the following command sequence:

```
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$ █
```

# catkin\_make

---

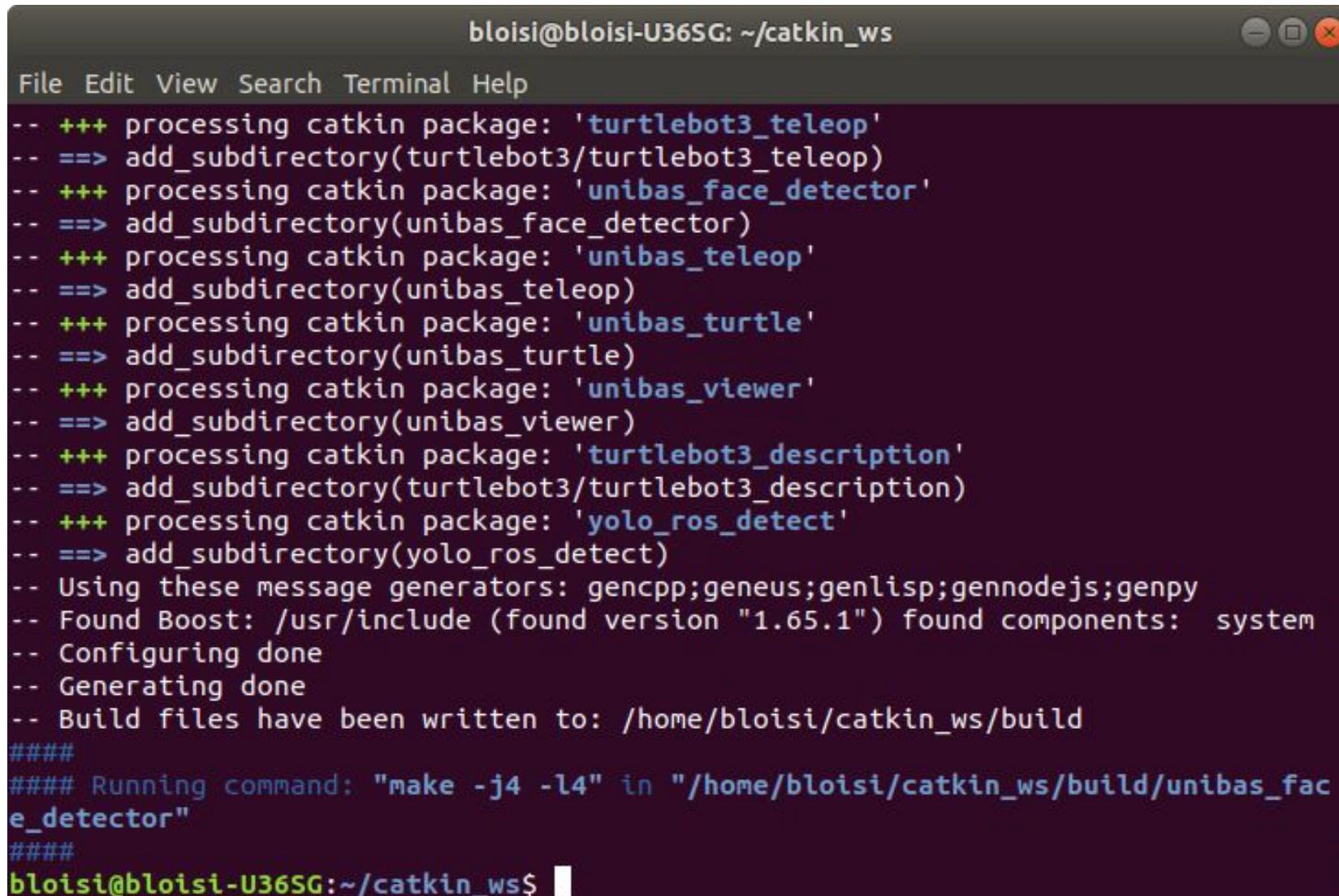
```
bloisi@bloisi-U36SG: ~/catkin_ws
File Edit View Search Terminal Help
bloisi@bloisi-U36SG:~/catkin_ws/src$ cd ..
bloisi@bloisi-U36SG:~/catkin_ws$ catkin_make --pkg unibas_face_detector
Base path: /home/bloisi/catkin_ws
Source space: /home/bloisi/catkin_ws/src
Build space: /home/bloisi/catkin_ws/build
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.
```

# unibas\_face\_detector installato

---



A screenshot of a terminal window titled "bloisi@bloisi-U36SG: ~/catkin\_ws". The window shows the output of a "catkin build" command. The text is color-coded: green for informational messages, blue for package names, and yellow for file paths. The output lists several packages being processed: "turtlebot3\_teleop", "unibas\_face\_detector", "unibas\_teleop", "unibas\_turtle", "unibas\_viewer", "turtlebot3\_description", and "yolo\_ros\_detect". It also mentions message generators like gencpp, geneus, genlisp, gennodejs, and genpy. Boost version 1.65.1 was found, and components system and boost were configured. The build files were written to "/home/bloisi/catkin\_ws/build". The command "make -j4 -l4" was run in the directory "/home/bloisi/catkin\_ws/build/unibas\_face\_detector". The terminal prompt at the bottom is "bloisi@bloisi-U36SG:~/catkin\_ws\$".

```
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

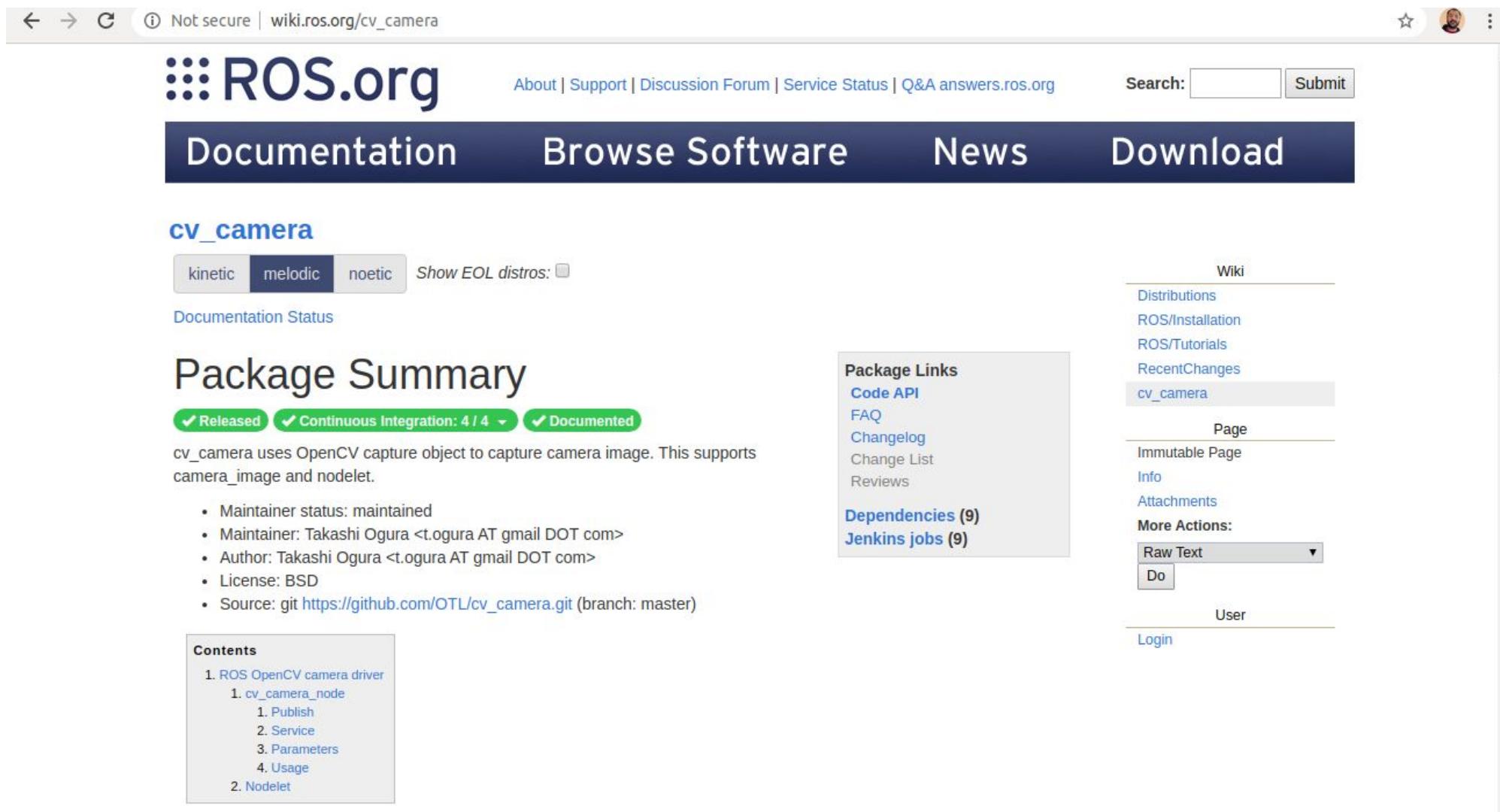
---

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. The main content area features a "Package Summary" with status indicators (Released, CI, Documented) and a brief description of the package. A "Contents" sidebar lists sections like ROS OpenCV camera driver, cv\_camera\_node, Publish, Service, Parameters, Usage, and Nodelet. To the right, there's a "Package Links" sidebar with links to Code API, FAQ, Changelog, Change List, and Reviews. Another sidebar shows "Dependencies (9)" and "Jenkins jobs (9)". On the far right, there's a "Wiki" sidebar with links to Distributions, ROS/Installation, ROS/Tutorials, RecentChanges, and cv\_camera (which is highlighted). Below that are "Page" and "User" sections with links for Immutable Page, Info, Attachments, More Actions (Raw Text, Do), and Login.

Not secure | wiki.ros.org/cv\_camera

ROS.org

About | Support | Discussion Forum | Service Status | Q&A answers.ros.org

Search:  Submit

Documentation    Browse Software    News    Download

**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**

1. ROS OpenCV camera driver
  1. cv\_camera\_node
    1. Publish
    2. Service
    3. Parameters
    4. Usage
  2. Nodelet

**Package Links**

[Code API](#)  
[FAQ](#)  
[Changelog](#)  
[Change List](#)  
[Reviews](#)

**Dependencies (9)**

[Jenkins jobs \(9\)](#)

**Wiki**

Distributions  
ROS/Installation  
ROS/Tutorials  
RecentChanges  
**cv\_camera**

**Page**

Immutable Page  
Info  
Attachments

**More Actions:**

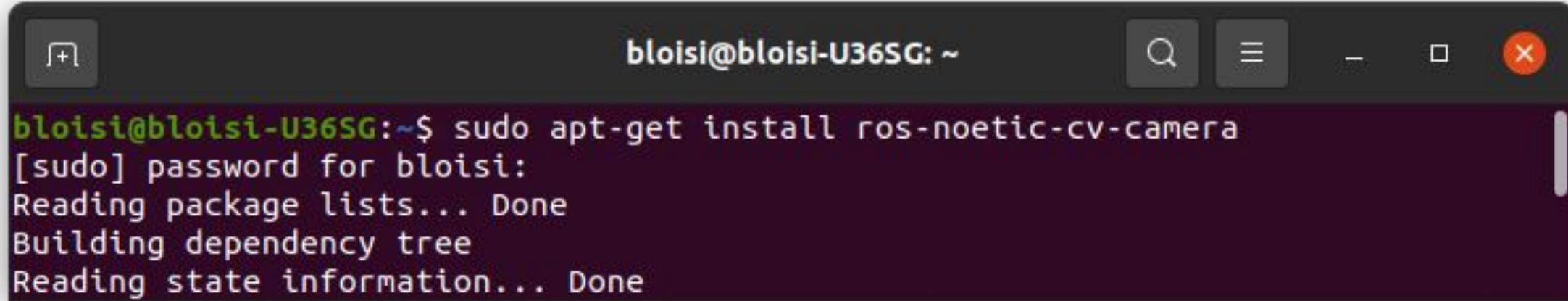
Raw Text ▾  
Do

**User**

Login

# installazione package cv\_camera

---



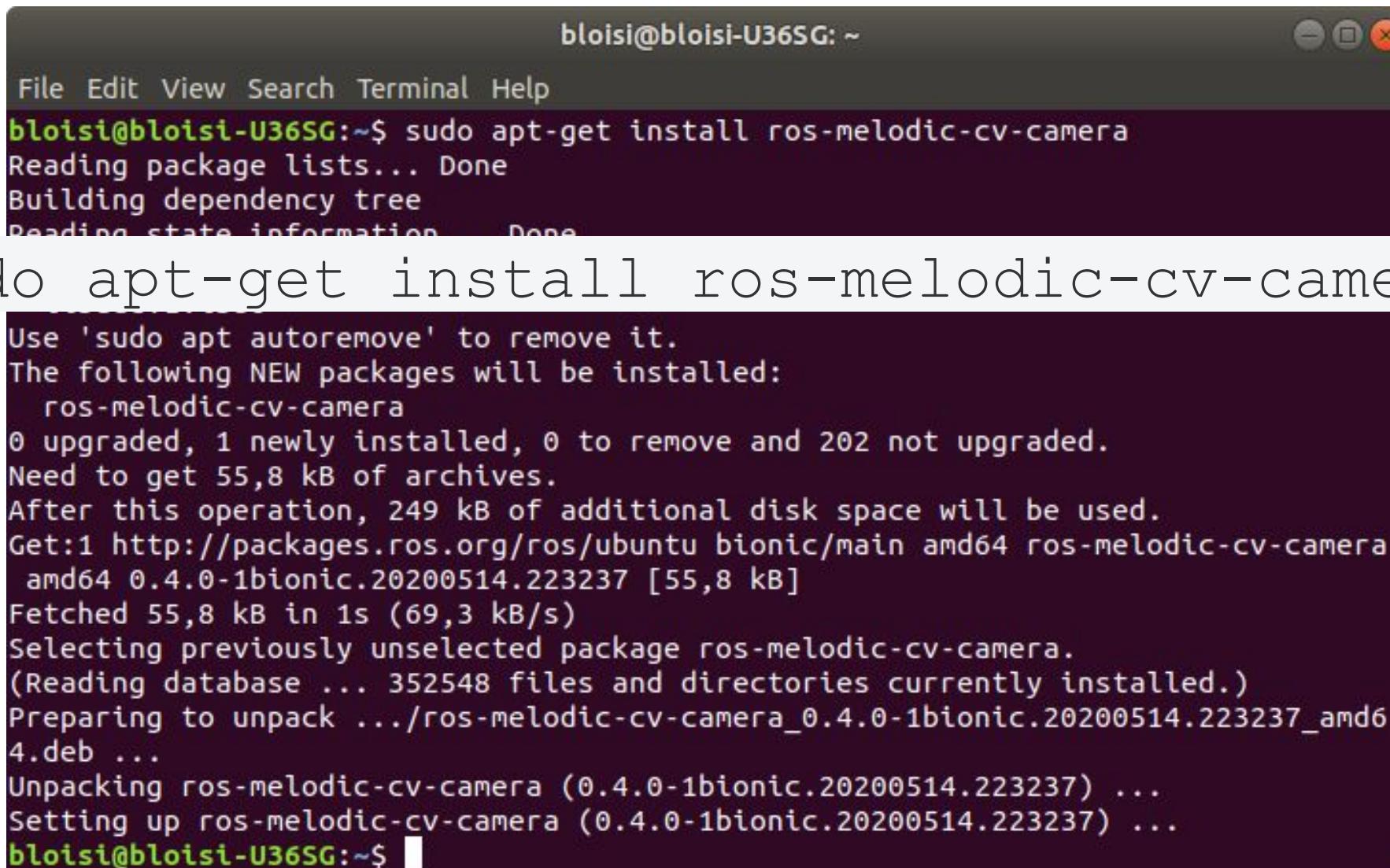
A screenshot of a terminal window titled "bloisi@bloisi-U36SG: ~". The window contains the following text:

```
bloisi@bloisi-U36SG:~$ sudo apt-get install ros-noetic-cv-camera
[sudo] password for bloisi:
Reading package lists... Done
Building dependency tree
Reading state information... Done
```

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

# package cv\_camera (melodic)

---

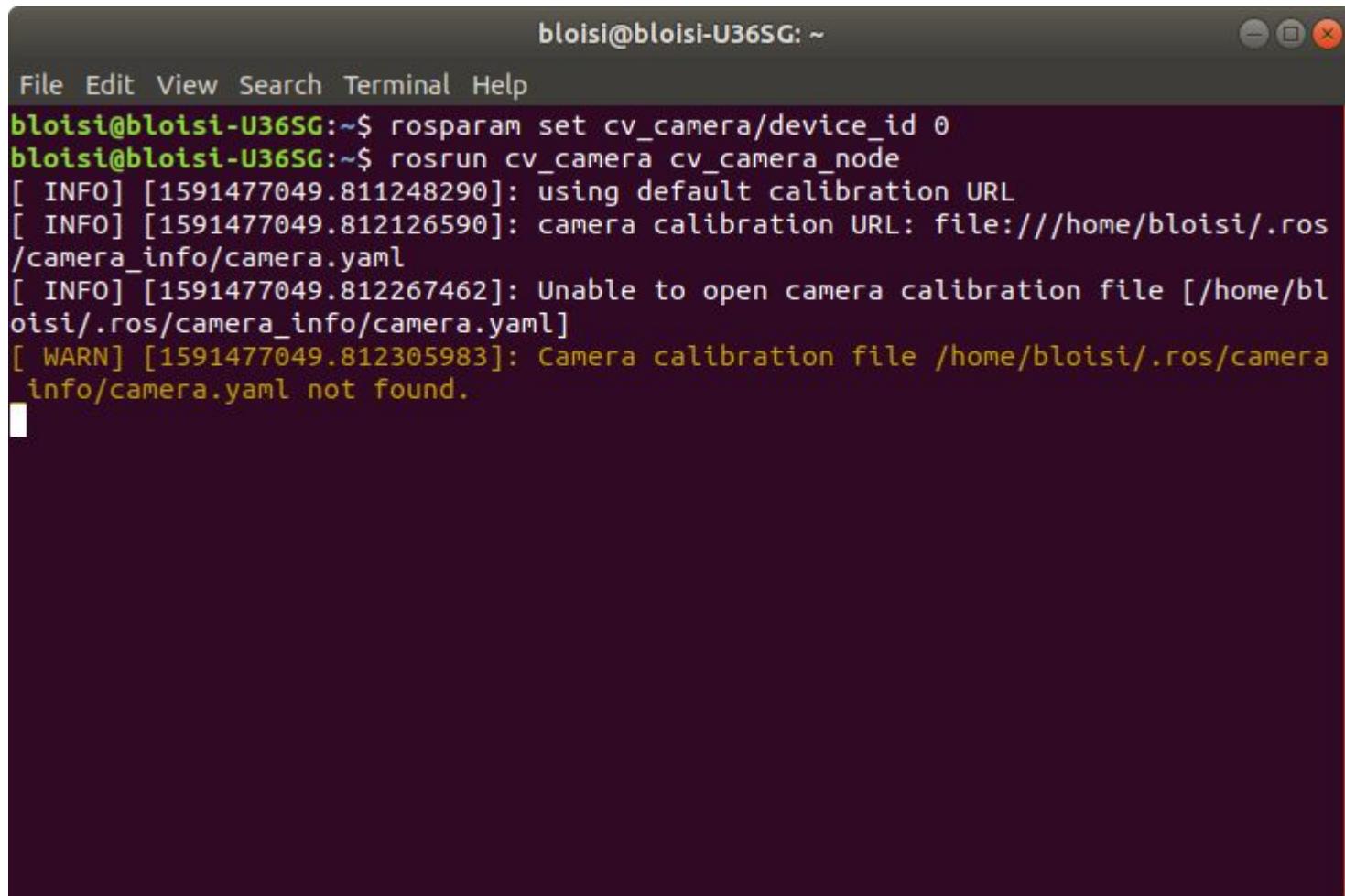


A screenshot of a terminal window titled "bloisi@bloisi-U36SG: ~". The window has a dark theme with white text. The terminal shows the command "sudo apt-get install ros-melodic-cv-camera" being run, followed by the output of the package manager. The output includes the following text:

```
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

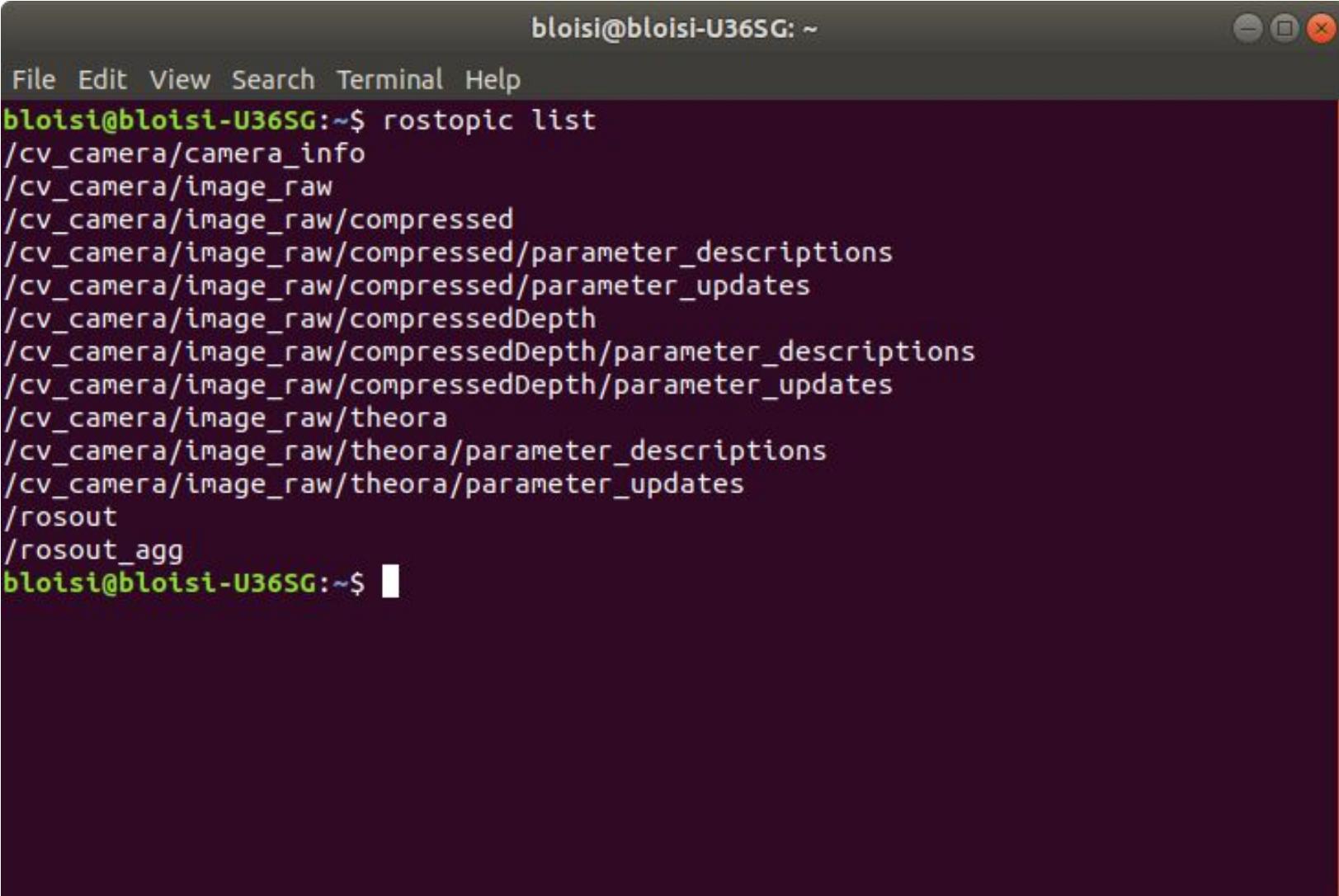
---

A screenshot of a terminal window titled "bloisi@bloisi-U36SG: ~". The window has a dark background with light-colored text. At the top, there's a menu bar with "File", "Edit", "View", "Search", "Terminal", and "Help". The main area of the terminal shows the following command-line session:

```
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.
```

# scopriamo i topic di cv\_camera

---



A screenshot of a terminal window titled "bloisi@bloisi-U36SG: ~". The window shows the output of the command "rostopic list", which lists various ROS topics. The topics listed are:

```
File Edit View Search Terminal Help
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

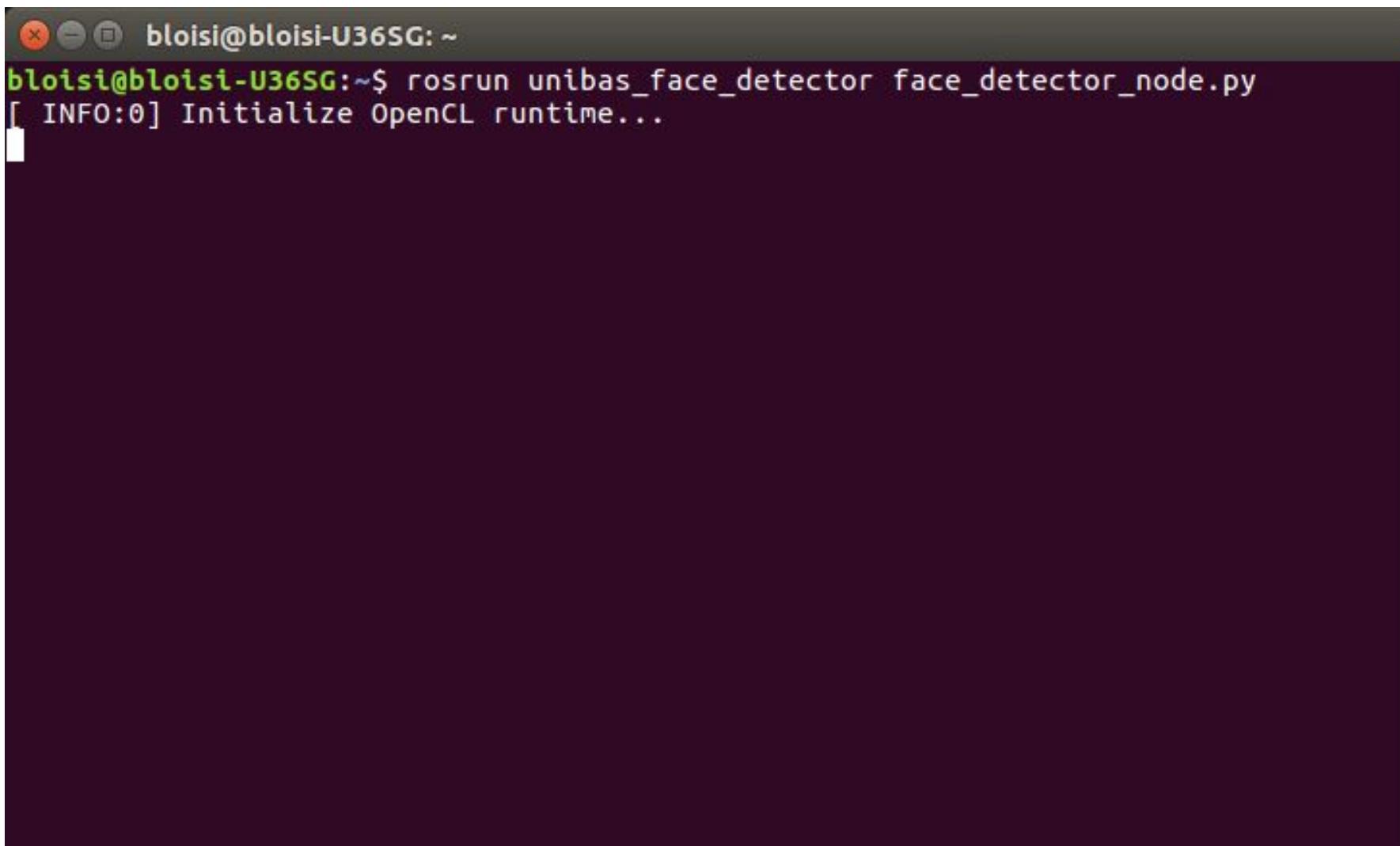
---

```
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 oval 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, the current line is 20, column 51, and the mode is INS.

# lanciamo il nodo face\_detector\_node

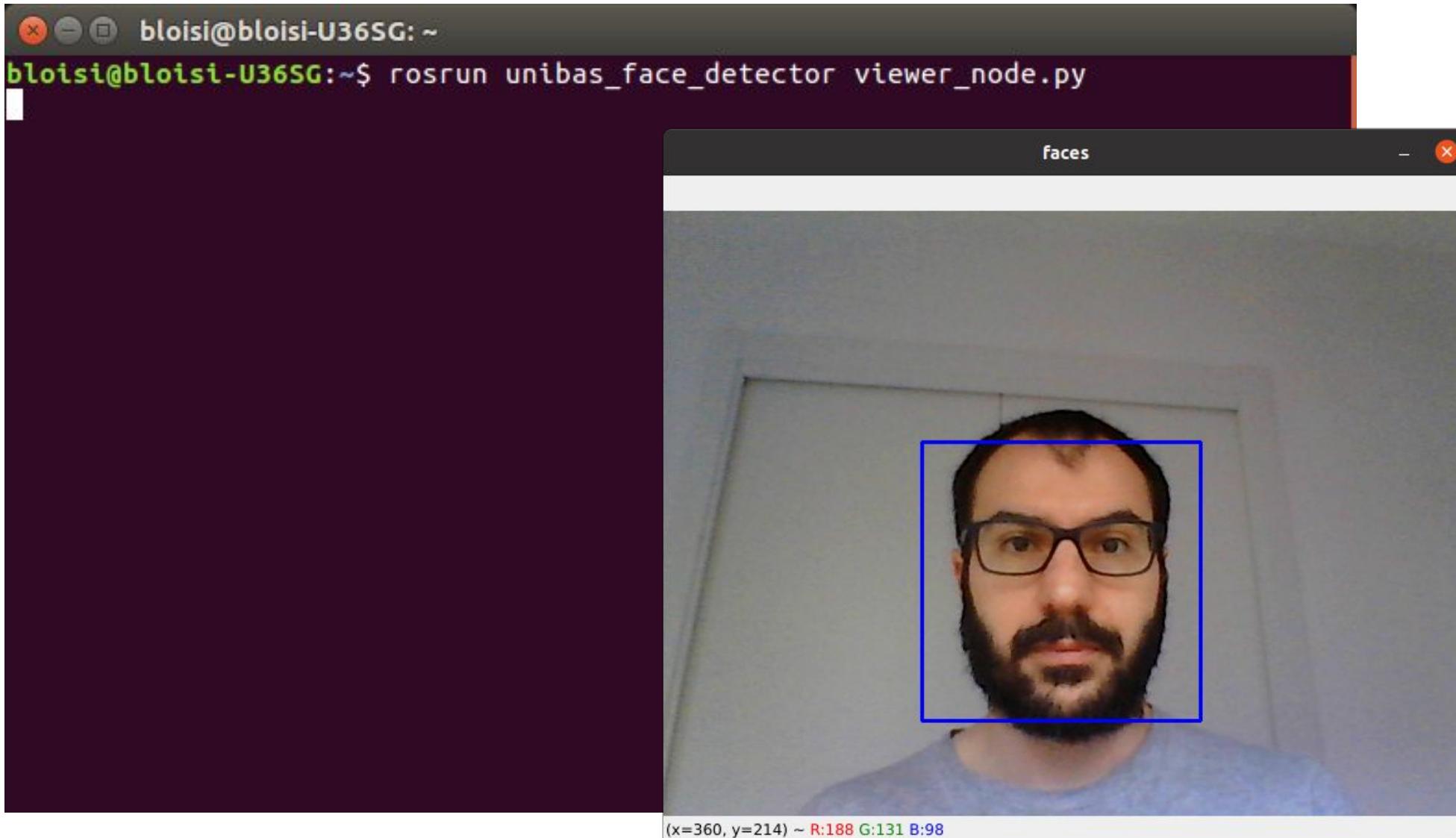
---



```
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>

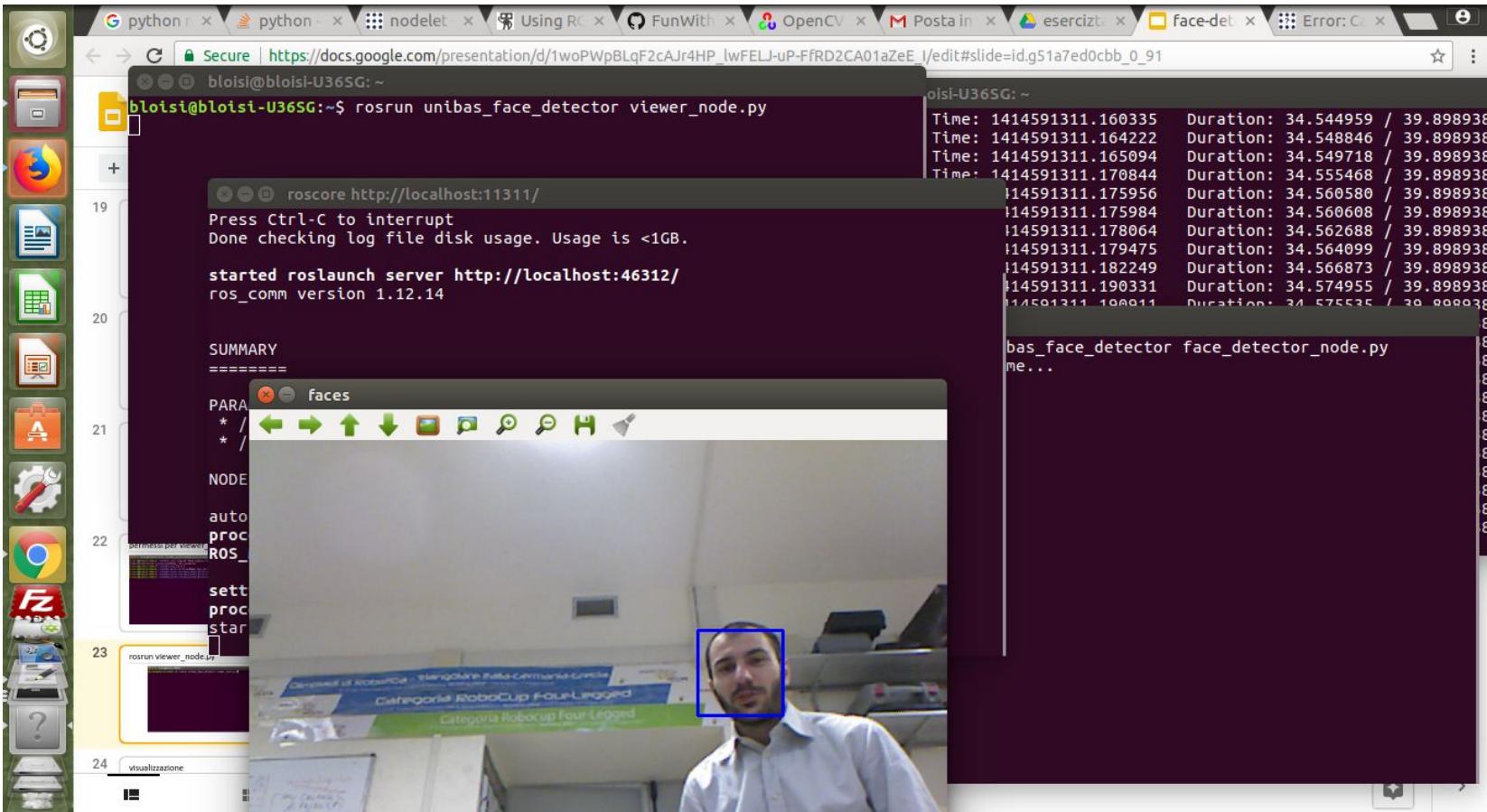
con il package unibas\_face\_detector

La detection dei volti presenti nella scena è corretta?

# Esercizio 2

---

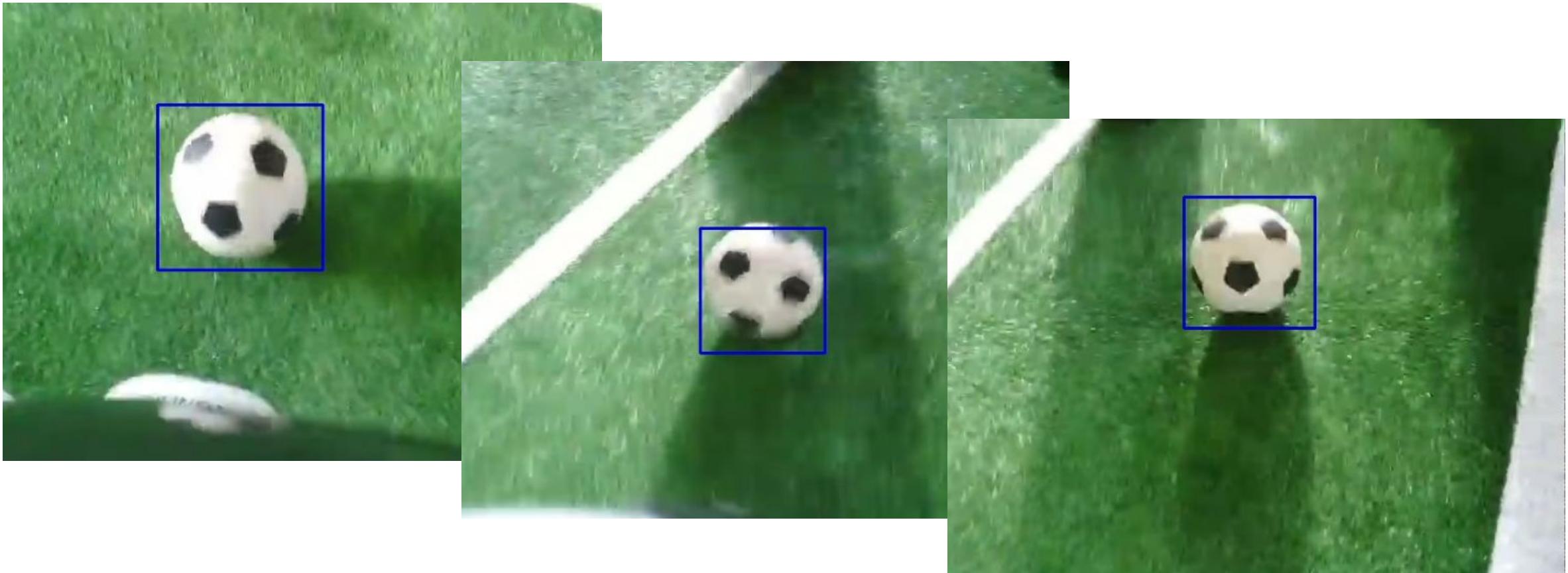
Creare un ros launchfile per evitare di dover aprire quattro differenti terminal per utilizzare il package `unibas_face_detector`



# Esempio Ball Detector

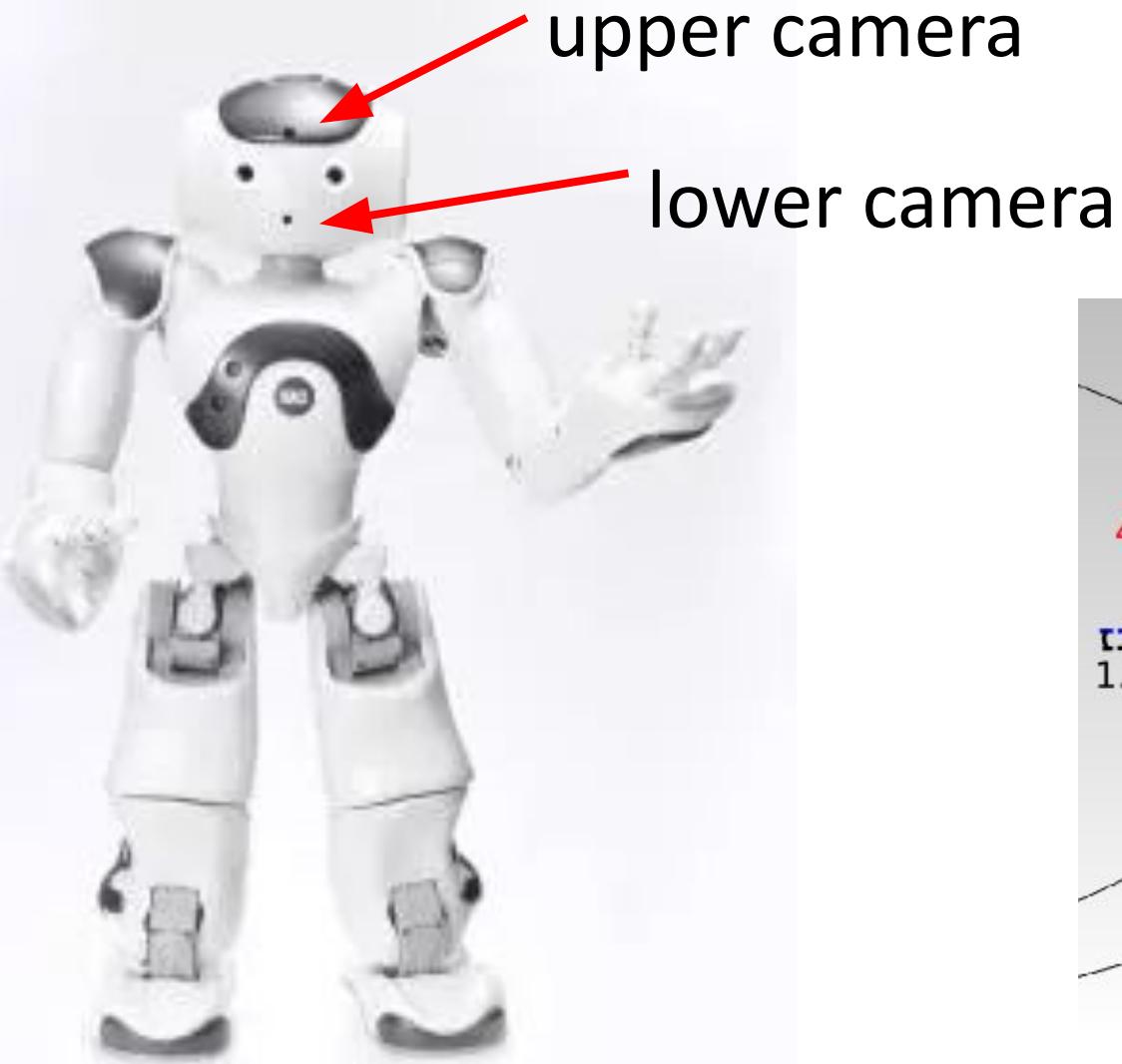
---

Provare a creare un ball detector per i robot NAO che giocano a calcio



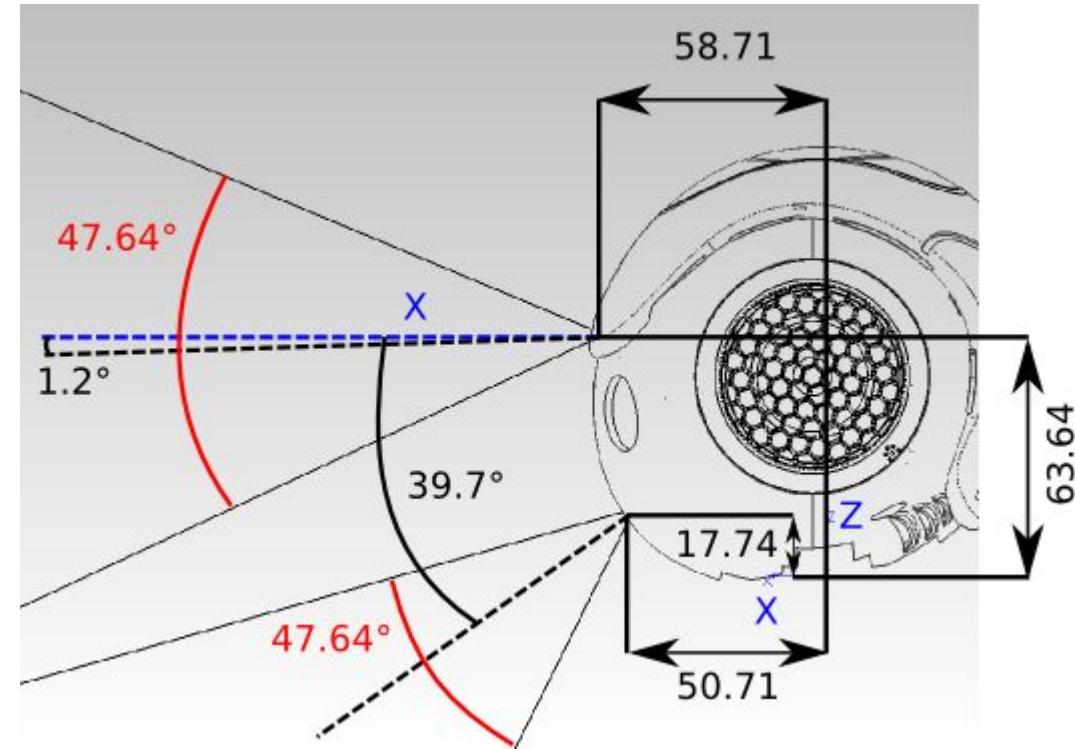
# NAO cameras

---



upper camera

lower camera



# Ball detector

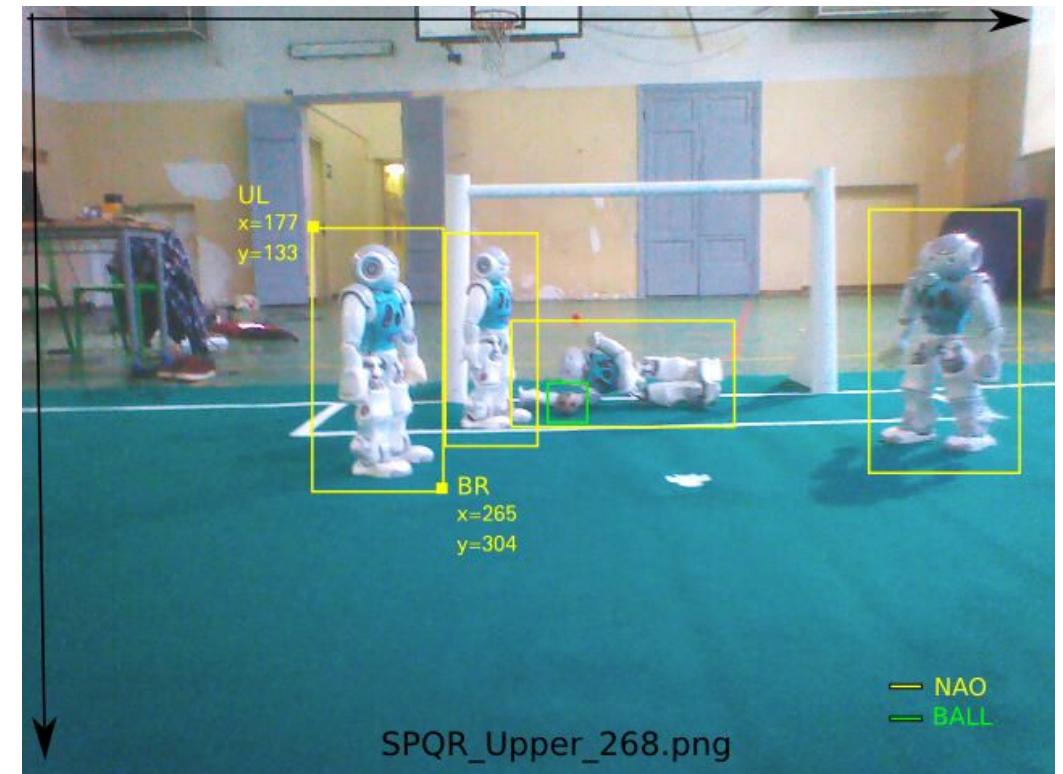
---

Bottom camera cascade classifier available at

[http://www.diag.uniroma1.it/%7Espqr/SPQR-NAO-image-dataset/bottom\\_cascade.xml](http://www.diag.uniroma1.it/%7Espqr/SPQR-NAO-image-dataset/bottom_cascade.xml)

## SPQR Team NAO image dataset

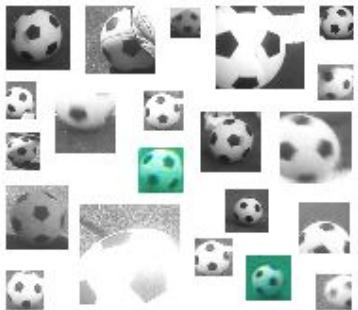
<http://www.diag.uniroma1.it/~labrococo/?q=node/459>





# How to Use OpenCV for Ball Detection

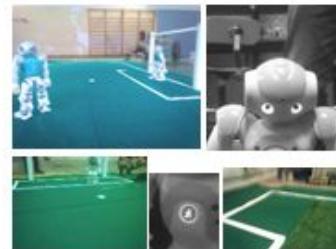
RoboCup SPL Use Case



## Positive set

It is crucial to generate accurately the positive set to obtain good results. We need to select image patch containing the object of interest trying to mantain the same proportion for each selected patch. To this end, we can use a graphical tool to crop the images. [Cropping](#) is a tool for creating image patches that can be used for machine learning with Haar or LBP based classifiers. [Cropping](#)

allows to select with the mouse and to save image patches with a constant proportion, which is crucial for generating a good positive sample set.



## Negative set

The images that are included in the negative set do not have to contain the object that we want to detect. If we know in advance which will be the application scenario for our detector, then it could be good to use images taken from it. The number of negative samples to use should be bigger than the number of positive samples. Generally, a proportion of 2:1 between negative and positive samples is considered acceptable, i.e., if the number of positive samples is 2,000 then the negative samples should be at least 4,000.

# Bag

---

scaricabile al seguente indirizzo

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

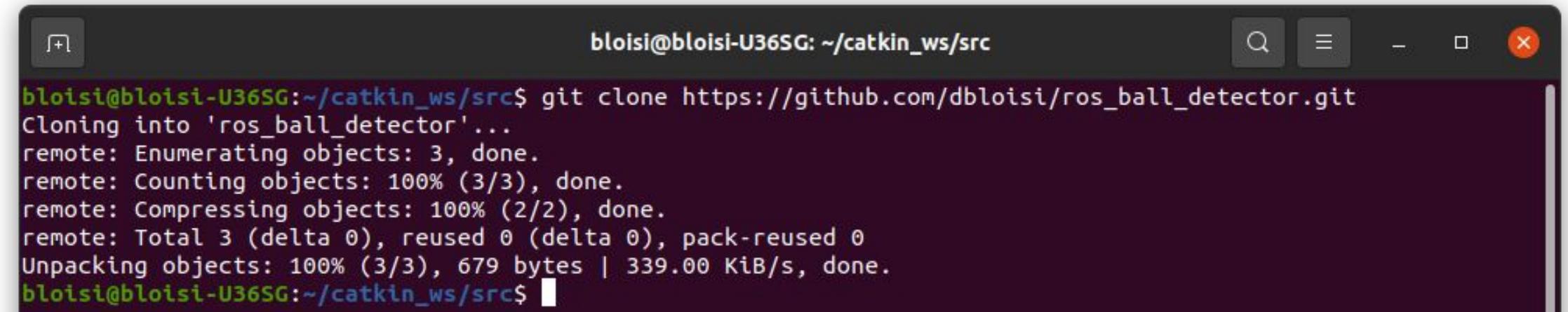
\* la bag di test è stata creata a partire dal video YouTube

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



# Code

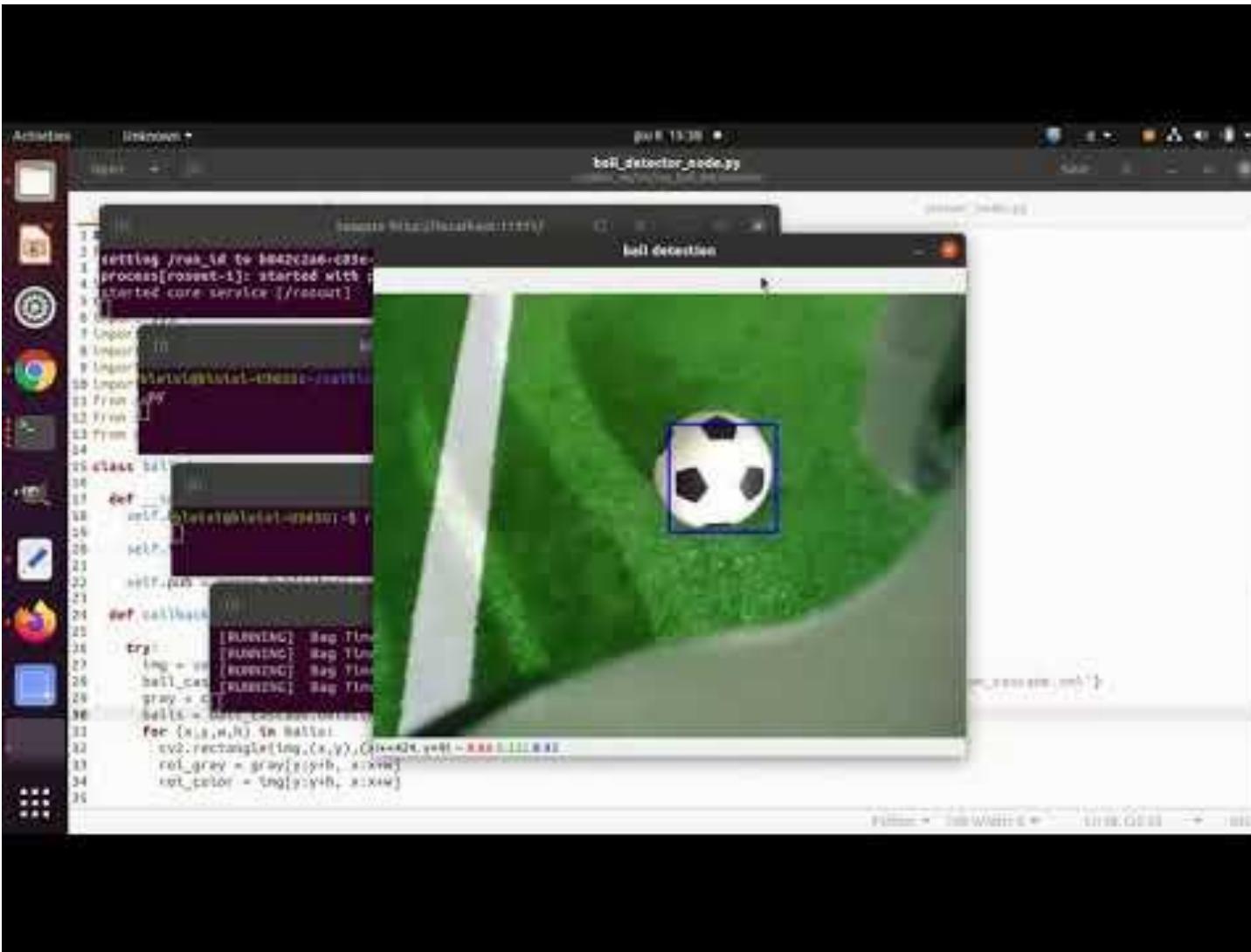
---



A screenshot of a terminal window titled "bloisi@bloisi-U36SG: ~/catkin\_ws/src". The window contains the following text output from a "git clone" command:

```
bloisi@bloisi-U36SG:~/catkin_ws/src$ git clone https://github.com/dbloisi/ros_ball_detector.git
Cloning into 'ros_ball_detector'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 679 bytes | 339.00 KiB/s, done.
bloisi@bloisi-U36SG:~/catkin_ws/src$
```

# Results



<https://youtu.be/sOPDA7kbu-o>

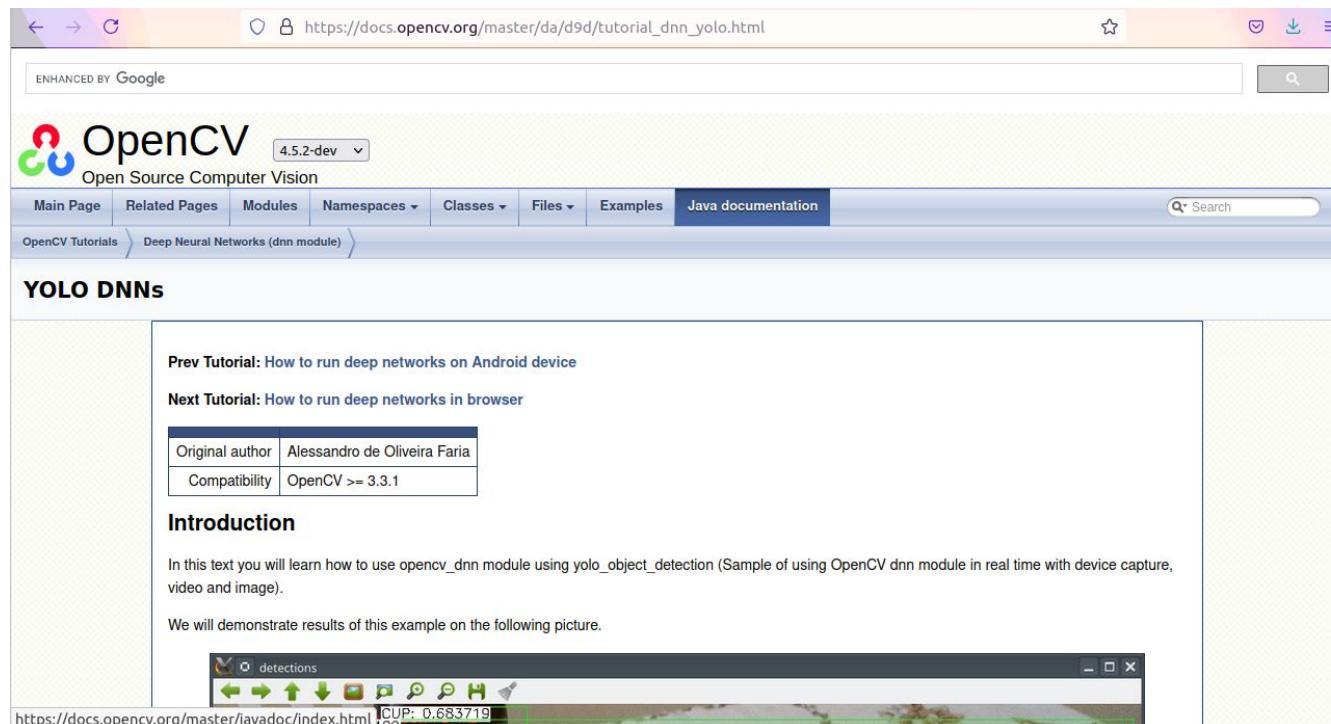
# Esercizio YOLO

---

Modificare il codice disponibile qui

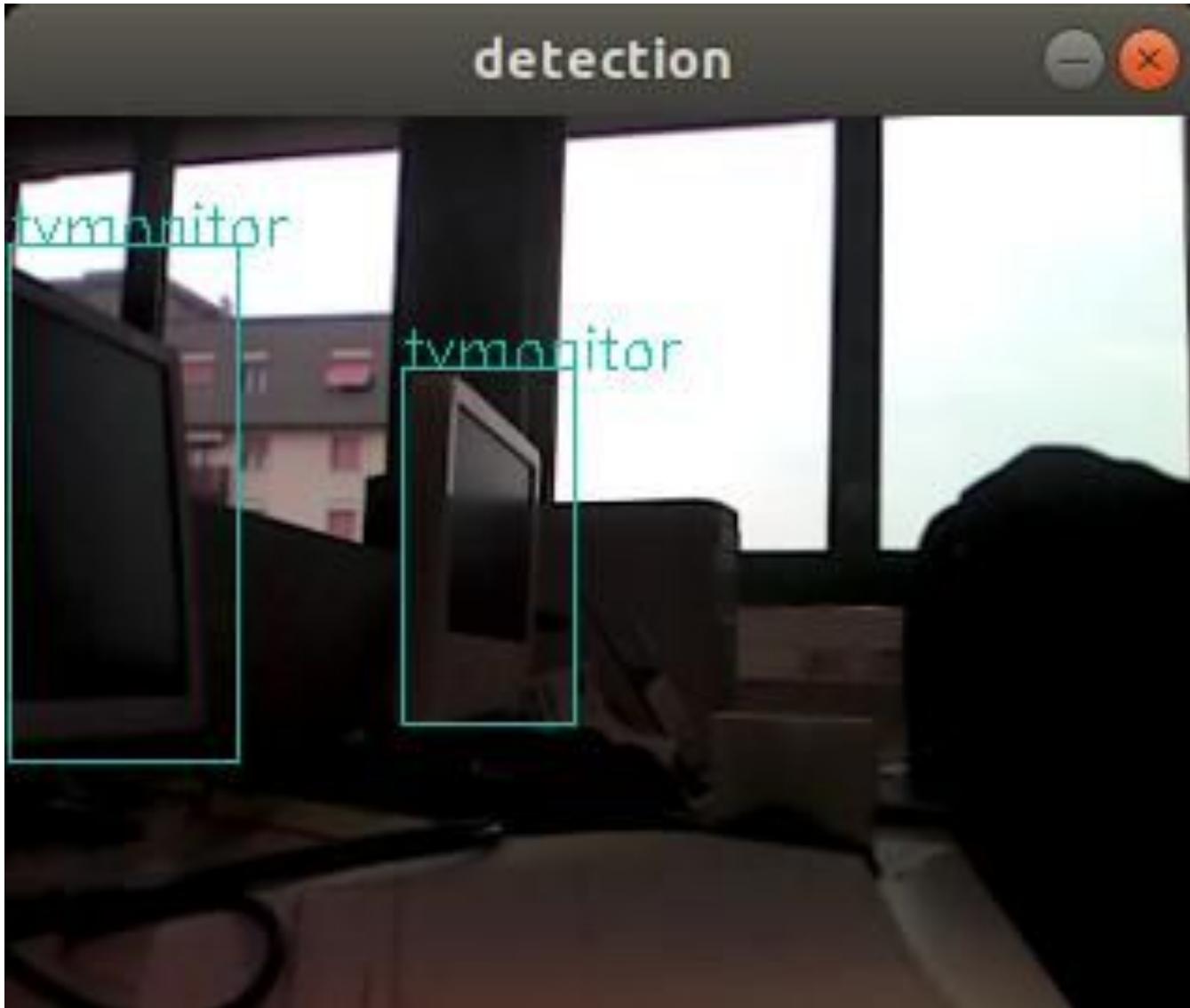
[https://docs.opencv.org/master/da/d9d/tutorial\\_dnn\\_yolo.html](https://docs.opencv.org/master/da/d9d/tutorial_dnn_yolo.html)

per creare un package ROS



# Esempio YOLO

---





**UNIVERSITÀ DEGLI STUDI  
DELLA BASILICATA**

# *Corso di Visione e Percezione*

# Esempi detection

