EXTERNSHIP PROJECT

Face Detection Robot Simulation Using ROS

Team Members

Divyansh Agarwal
Dishika Poddar
Abin Gireesh
Rajat Srivastava

Introduction

- OpenCV (Open Source Computer Vision Library) is a library of programming functions mainly aimed at real-time computer vision.
- The library is cross-platform and free for use under the open-source Apache 2 License. Starting with 2011, OpenCV features GPU acceleration for real-time operations.

Goals to achieved using this software

- Advance vision research by providing not only open but also optimized code for basic vision infrastructure. No more reinventing the wheel.
- Disseminate vision knowledge by providing a common infrastructure that developers could build on, so that code would be more readily readable and transferable.
- Advance vision-based commercial applications by making portable, performance-optimized code available for free – with a license that did not require code to be open or free itself.

Problem Statement

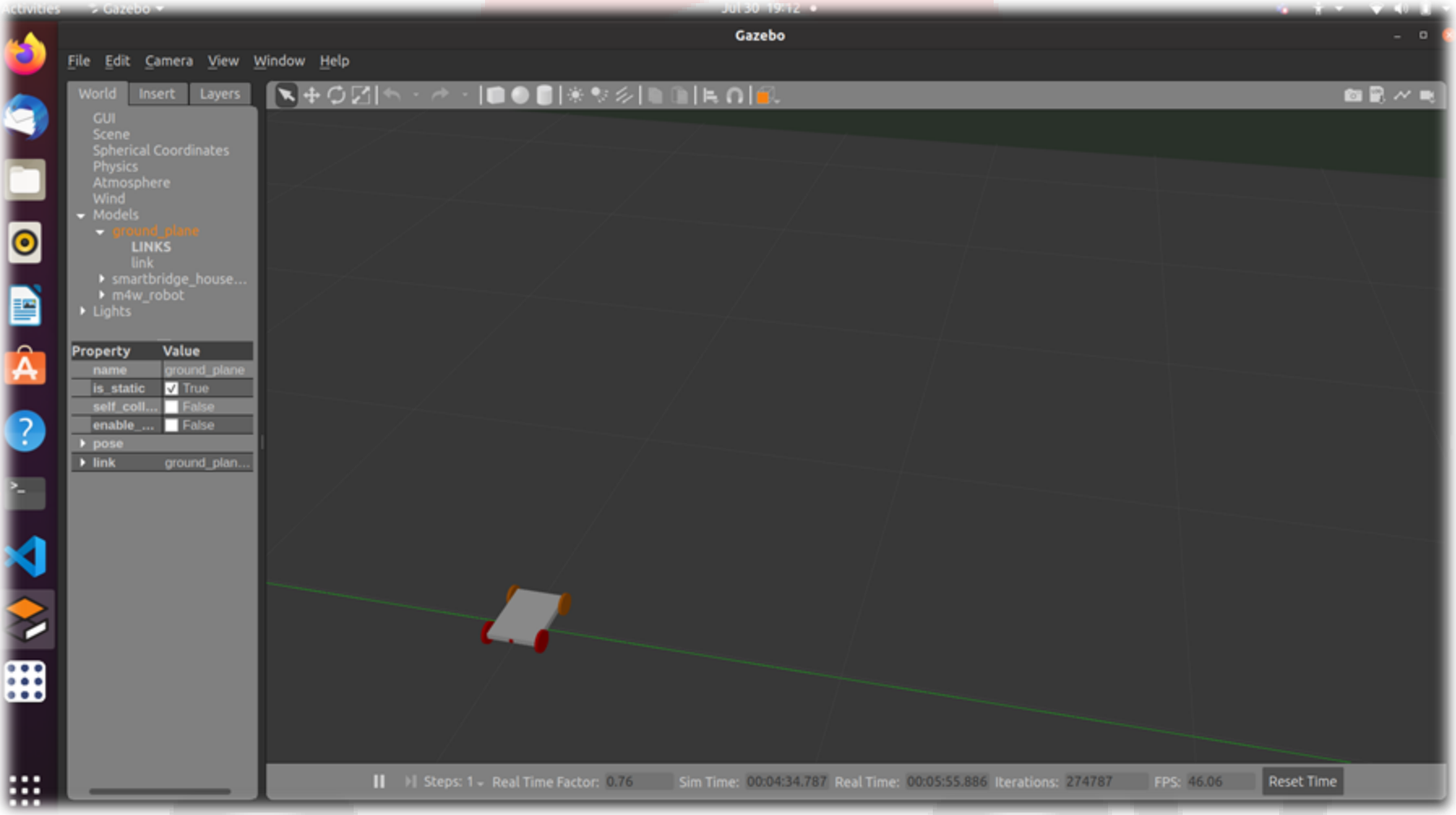
A face-based user verification robot serves many purposes in places like a multi-specialty hospital with tens of doctors and hundreds of patients. In places like that, directing the patients to the concerned doctor is a difficult and risky task for a human to do as the diseases are contagious. So, a face-based user verification robot checks for the scheduled appointment of the patient with the respective doctor and navigates them to the particular chamber without human intervention.

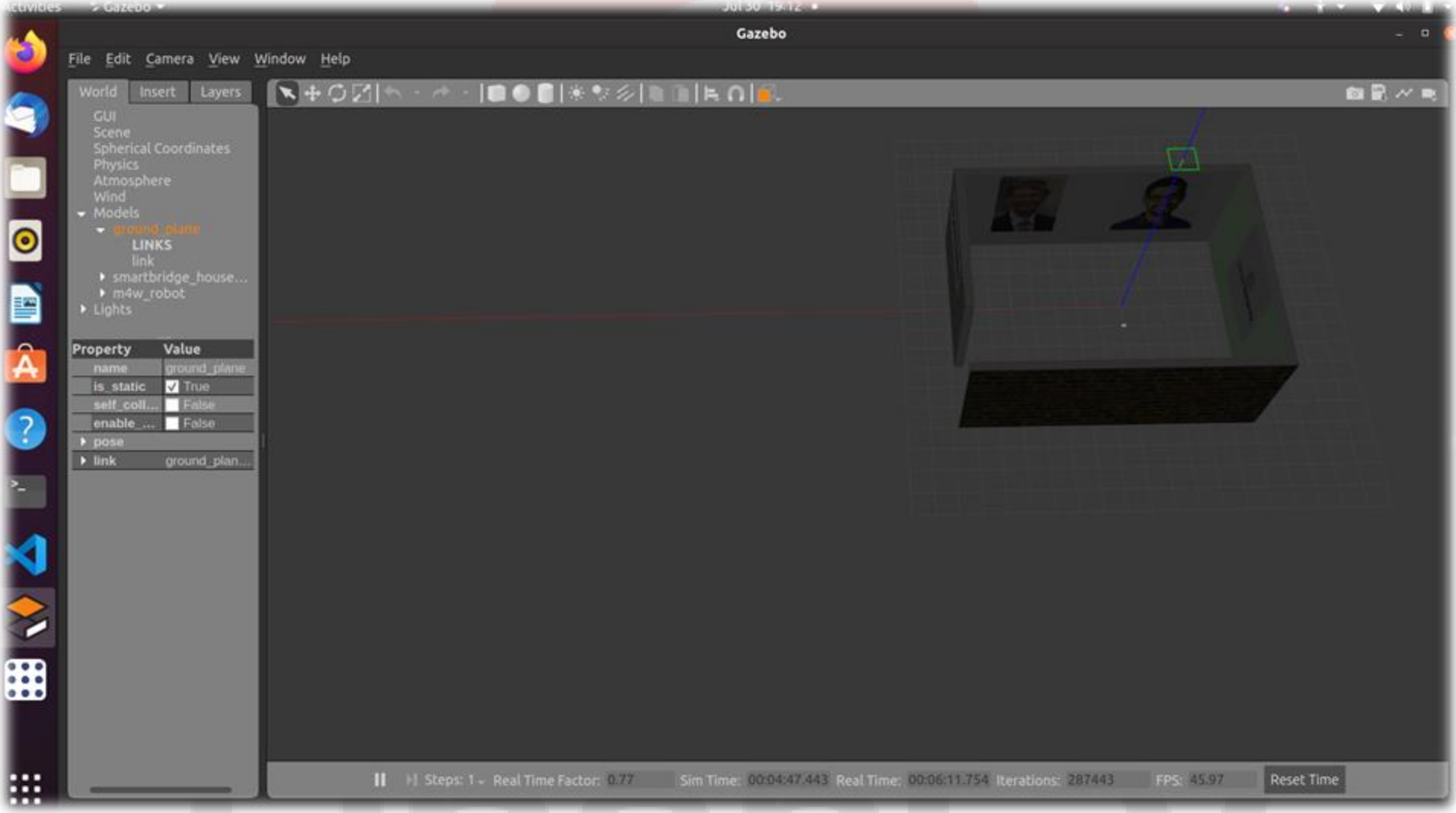
Programming languages

- Open CV is written in C++ and its primary interface is in C++, but it still retains a less comprehensive though extensive older C interface. All of the new developments and algorithms appear in the C++ interface. There are bindings in Python, Java and MATLAB/OCTAVE. The API for these interfaces can be found in the online documentation. Wrappers in several programming languages have been developed to encourage adoption by a wider audience. In version 3.4, JavaScript bindings for a selected subset of OpenCV functions was released as OpenCV.js, to be used for web platforms.

Description

The face detector employs the OpenCV face detector (based on a cascade of Haar-like features) to obtain an initial set of detections. It then prunes false positives using stereo depth information. The depth information is used to predict the real-world size of the detected face, which is then preserved as a true face detection only if the size is realistic for a human face. This removes the majority of false positives given by the OpenCV detector. This module was deployed on a gazebo robot built with help of URDF and XACRO files.


Gazebo
File Edit Camera View Window Help
World Insert Layers
GUI
Scene
Spherical Coordinates
Physics
Atmosphere
Wind
Models
ground_plane
LINKS
link
smartbridge_house...
m4w_robot
Lights
Property Value
name ground_plane
is_static True
self_coll... False
enable_... False
pose
link ground_plan...
Steps: 1 Real Time Factor: 0.76 Sim Time: 00:04:34.787 Real Time: 00:05:55.886 Iterations: 274787 FPS: 46.06
Reset Time


Activities
Gazebo
Jul 30 19:12
Gazebo
File Edit Camera View Window Help
World Insert Layers
GUI
Scene
Spherical Coordinates
Physics
Atmosphere
Wind
Models
ground_plane
LINKS
link
smartbridge_house...
m4w_robot
Lights
Property Value
name ground_plane
is_static True
self_coll... False
enable_... False
pose
link ground_plan...
Steps: 1 Real Time Factor: 0.77 Sim Time: 00:04:47.443 Real Time: 00:06:11.754 Iterations: 287443 FPS: 45.97
Reset Time

Advantages

- ROS and OpenCV open sourced
- Prebuilt packages and Reusability
- Low cost and Time saving
- Active community and documentation
- Regular updates

Disadvantages

- ROS is complex
- Multiple distribution/versions
- Not on major Operating Systems
- Backward compatibility

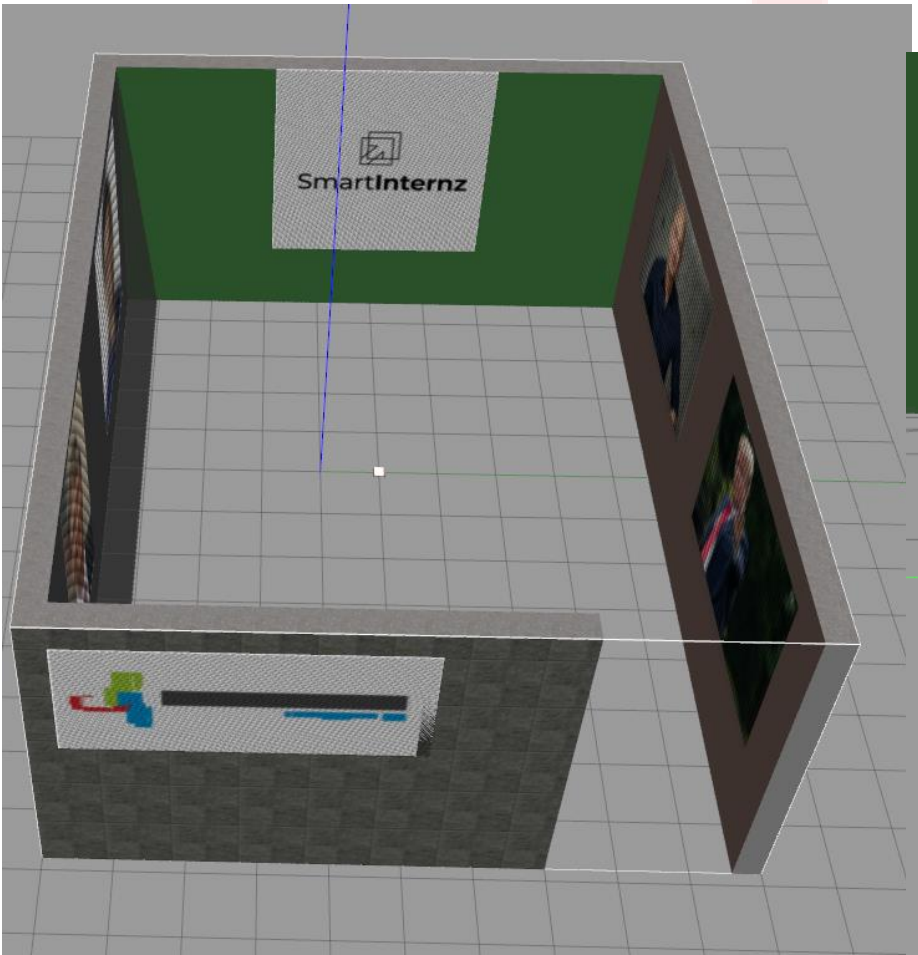
APPLICATIONS

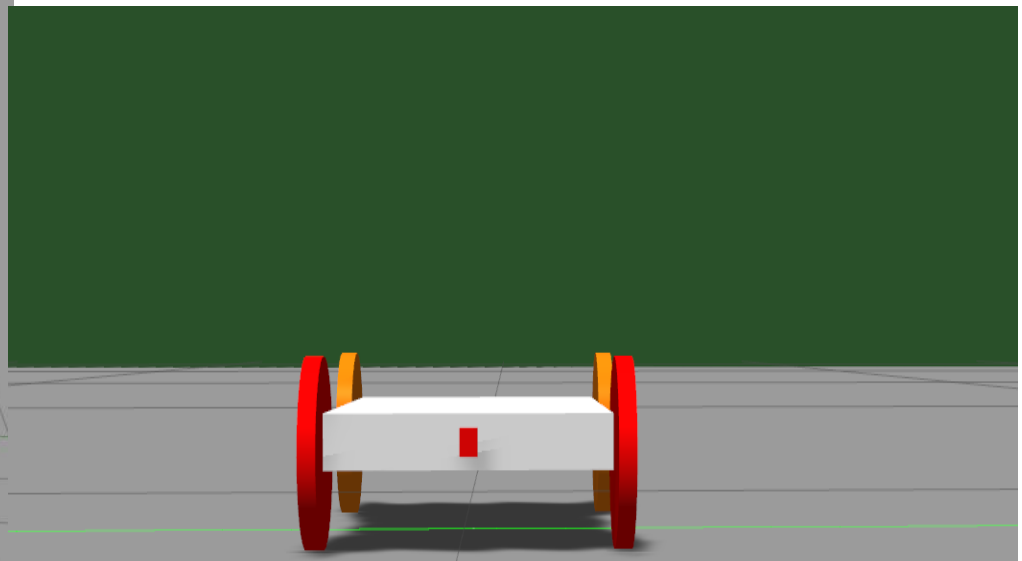
- A face-based user verification robot serves many purposes in places like a multi-specialty hospital with tens of doctors and hundreds of patients. In places like that, directing the patients to the concerned doctor is a difficult and risky task for a human to do as the diseases are contagious. So, a face-based user verification robot checks for the scheduled appointment of the patient with the respective doctor and navigates them to the particular chamber without human intervention
- Especially during this pandemic this bot will be extremely useful in reducing human contact.
- It can be used in places such as banks and offices as well, to escort people while maintaining a safe distance from them.
- It can be used at old age homes and schools, it reduces their exposure to human beings which is quite necessary since they are the most vulnerable population for the upcoming third wave of the novel coronavirus.

FUTURE SCOPE

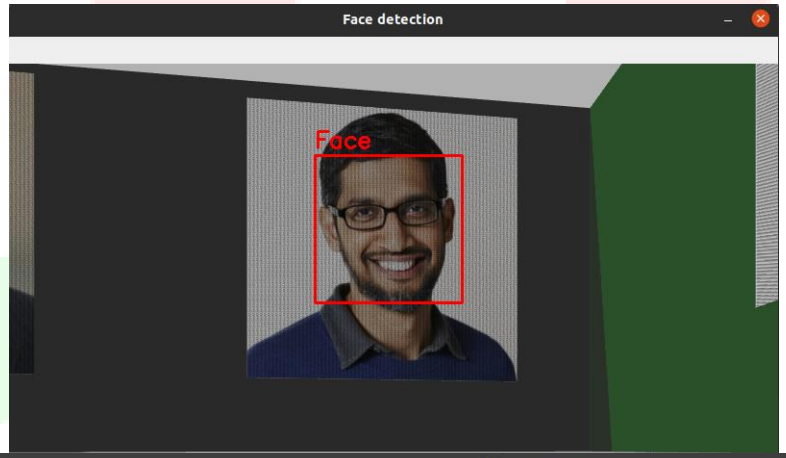
This bot can be made absolutely autonomous which would completely eradicate the need for human intervention. It can be integrated with a image recognition model that can identify humans who are unknown and have features such as buzzers or other forms of alarm and a web interface that warns the user of the trespasser on their property. Hence it can act as a huge asset in places that need high security while keeping humans out of the risk.

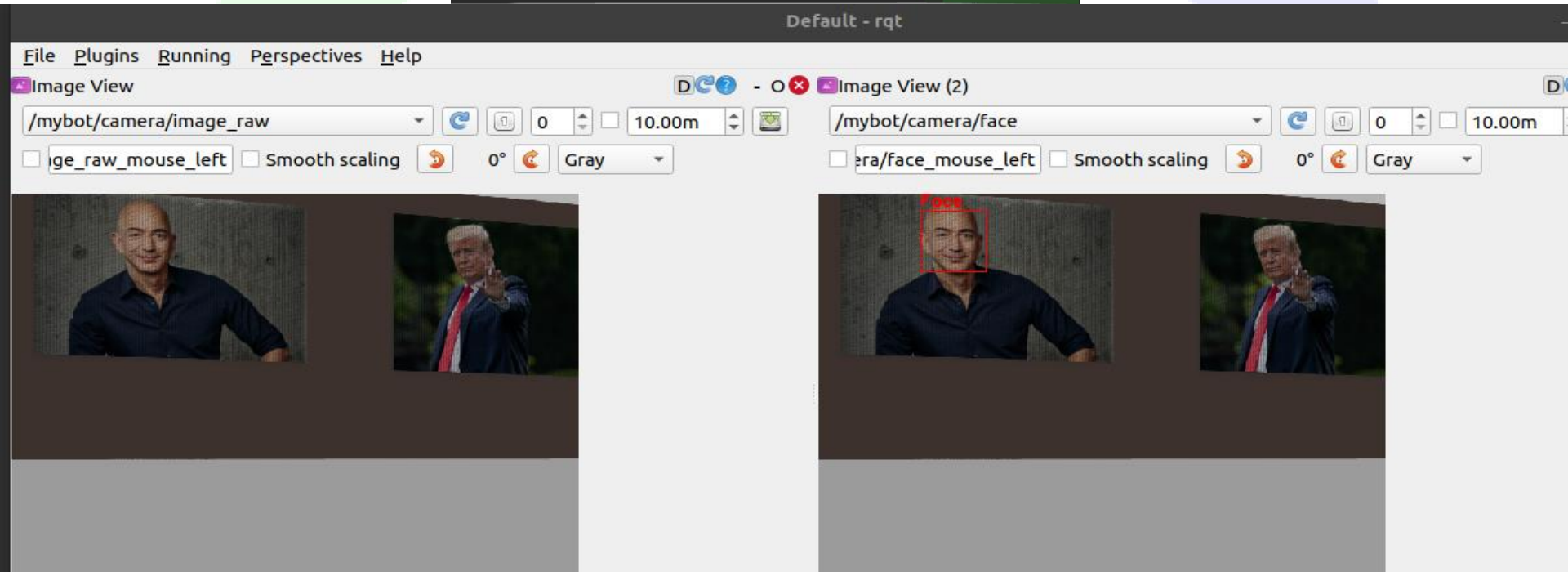
Project



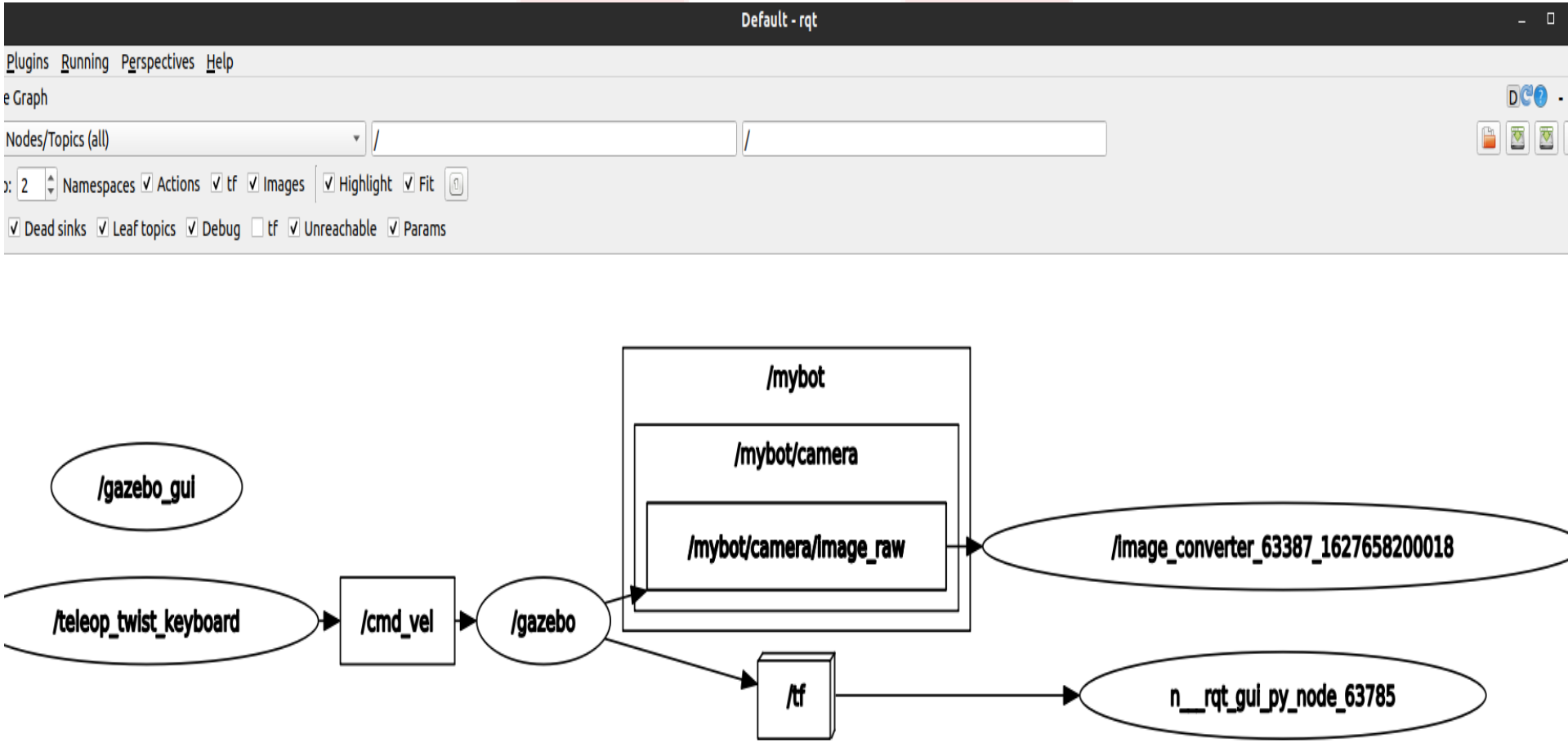


Face detection





Graph


Default - rqt
Plugins Running Perspectives Help
e Graph
Nodes/Topics (all)
/
/
Namespaces
Actions
tf
Images
Highlight
Fit
Dead sinks
Leaf topics
Debug
tf
Unreachable
Params
/mybot
/mybot/camera
/mybot/camera/image_raw
/gazebo_gui
/teleop_twist_keyboard
/cmd_vel
/gazebo
/tf
/image_converter_63387_1627658200018
n__rqt_gui_py_node_63785

Conclusion

In this project we have successfully deployed Open CV and with the ROS packages we have developed a Face detecting robot which can detect faces, and wish to further explore and incorporate how "Computer Vision" can be incorporated in various other fields of Robotics.