

Module name: Object classification

- **Main functionalities:**

A deep convolutional neural network (CNN) is used to classify and sort objects. When industrial robot picks the object, it is then classified using Convolutional Neural Network. This module allows for sorting different types of objects, for example as can be seen in Fig.4 and Fig.5 two different kinds of objects are being sorted. In order to train the classifier to recognize new classes of objects, new training datasets must be provided.

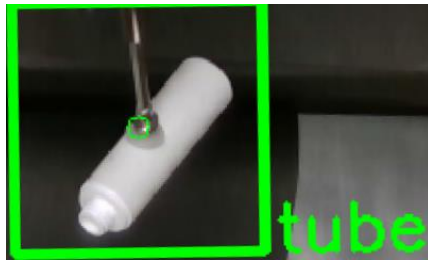


Fig. 2: Class tube



Fig. 1: Class can

- **Technical specifications:**

The overall system description is shown in Fig. 6.

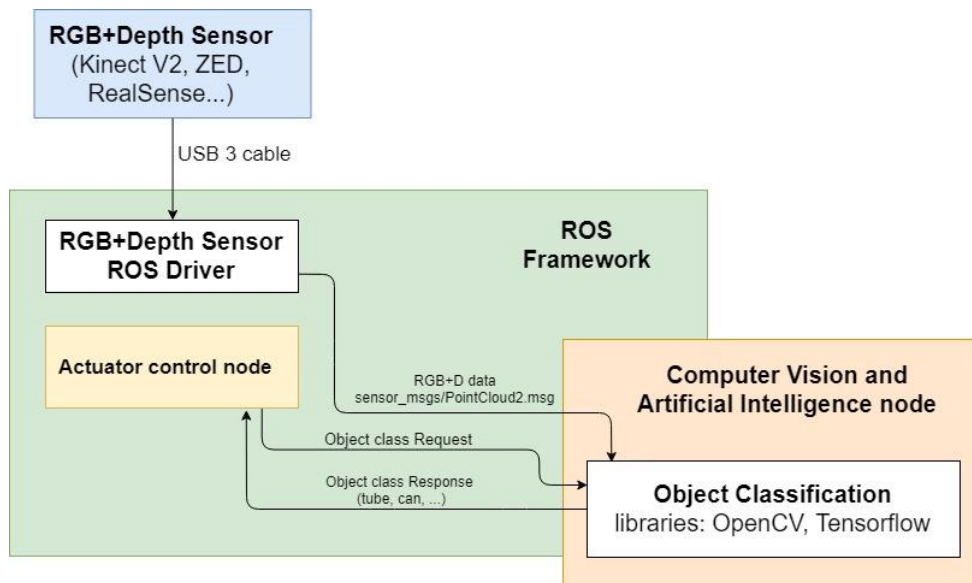


Fig. 3: Module components

Training can be done on standard desktop PC, to ensure precision up to 99% training model requires at least 1000 images of the object. The maximum amount of the different object classes is not specified, the system has been tested with 7 different types of classes.

The depth sensor is connected to PC that runs the ROS Kinetic on Ubuntu 16.04. Currently, Intel RealSense d415, d435, Kinect v2 depth cameras are supported, but any camera with ROS driver can be used, if the data can be published as PointCloud2. All the software for this module is implemented using Python 2.7 programming language.

- **Inputs and outputs:**

All the data is transferred via a standard ROS transport system with publish/subscribe and request/response semantics. This module subscribes to RGB+Depth sensor data and produces requested object class.

- **Interface specification:**

User can specify region where objects will be classified. The classified object type is shown in the RGB image.

- **Formats and standards used:**

ROS service communication to request object class.

The sensor data is received from the sensor driver in sensor_msgs.PointCloud2.msg format.

ROS, OpenCV, Tensorflow, PCL, Python standard libraries

- **Availability:**

Available, repository link can be provided by request.

- **Application scenarios:**

Object sorting.

- **Offered for internal / external use**

Internal and external