## General course objectives

3D object detection ground on deep learning is getting attention these years. Since LiDAR is quickly updated, deep learning makes self-driving cars possible. In the indoor scenario, RGB-D camera still plays an important role in 3D-scene understanding for indoor robots, because of the lower price and higher resolution. Even though high-performance GPUs provide the possibility to handle tons of data and to achieve high accuracy, deep neural networks with a large number of learnable parameters pose challenges to the real time of the detection algorithms, especially in robot applications. Well-designed neural networks are becoming the new hot spot in this promising field, since the improvements in hardware are harder and harder to make.

The purpose of the course is to give the students a glance into object detection in indoor scenario based on deep neural networks. Students are required to model a specific task and propose a solution and verify their method by code implementation. After the class, students are expected to get a further vision into the 3D object detection associated with deep learning.

## Learning objective

A student has met the object of the course will be able to:

* Understand the basic concepts on indoor 3D object detection
* Know the principle of 3D convolutional neural networks
* Design an efficient deep neural network
* Use popular open-source software library and toolbox to concretize the idea
* Keep code under version control with Git

## Content

Students are required to work on one of the most famous public datasets in indoor 3D object detection, which is called ScanNet and maintained by Stanford University. ScanNet is an RGB-D video dataset containing 2.5 million views in more than 1500 scans of 21 different object classes. The networks can be built on any popular deep learning software libraries, such as TensorFlow and PyTorch. Open-source toolbox may be of great help, e.g., MM Detection Toolbox developed by the MMLab in CUHK.

The course lasts for 3 weeks. In the first week, students need to get familiar with the datasets and toolbox. Ideas should be raised and determined no late than the beginning of the second week. In the second and third week, students should finish the coding works and deploy code to servers. They are required to submit a report in the last day of the third week.