**INTRODUCTION**

RGB-D object detection has become a popular research topic due to the demand from application such as automated car and LIDAR data processing. Recent progress of deep learning techniques has greatly improved model precision and processing time. In this project, we study one of the state-of-the-art 3D object detection frameworks - VoteNet, and try to improve it using RGB information.

**Method**

- The baseline VoteNet framework is shown at the bottom of the diagram. It only uses \((x, y, z)\) location of point cloud.
- We try to use 2D object detection as a hint to 3D object detection.
- We use 2D faster RCNN FPN to detect 2D bounding boxes \([xmin, ymin, xmax, ymax]\) and project them to 3D space to get a frustum. If a point from point cloud \(\{p_i\}_{i=1}^{n}\) is inside the frustum, then we assign the class score \(b_{i,j}\) to it. We combine the score feature with it’s 3D coordinate using

\[
g_i = p_i^t, b_{i,j}
\]

- We use SUN RGBD dataset for training and evaluation. The dataset contains 10335 scenes.

**Conclusion and Future Research**

Before we use a real 2D detector, we tried a proof of concept method using ground truth 2D bbox. Results are shown as VoteNet + 2D GT in the tables. Precision rate is improved by a lot.

Then we use a faster RCNN FPN to generate 2D bounding box on evaluation set. Results are shown as VoteNet + 2D. The precision and recall rate have dropped a lot.

From the point cloud image we can see that even in VoteNet + 2D GT there are a lot of false positive results. This could be a sign of over fitting.

To improve this, we can split training set into two parts, one for 2D detector training and one for 3D detector training. This will be our future work.
https://youtu.be/CBaVkr4Cv1o