**Introduction**

- When training a robot to perform tasks such as pick and place, a difficult step is to visually perceive the object and extract the localization information. Our goal in this project is to tackle this important task using synthesized RGB and depth images from the robot’s depth camera (eyes).

**Data Preprocessing**

- We explored the 1,800 images dataset generated by Unity3D simulation platform, provided by Stanford AI Lab and Deepmind. In this dataset, each sample is made of a 299x299x3 RGB image and a 299x299x1 depth image.
- On each image, there are 10-20 different items, and all of the items belong to a 55 items dataset. Currently, we are training the model to localize an old radio, shown in Figure 1(a) and (b).
- The dataset was processed using *domain randomization*, which adds different simulated light sources and hues to the generated images. As such, our model can better fit to images take in real-life situations in the future.
- In one of our models, we processed the depth image using *color map*, to convert it into RGB (299x299x3) image.

![Figure 1](image1.png)

**Models**

- **Model 1: 2D VGG Model**. We input a 299x299x3 image file, run through a convolution layer with filter size $= 3$, add padding and reshape to feed it into the Keras VGG16 model. After VGG16, we added a few more fully-connected layers to generate an output of estimated $(x, y, z)$ location of the target object.

![Model 1 Diagram](image2.png)

- **Model 2: 3D Voxel + VGG Model**. We preprocess the depth information to produce a spatial 3D voxel representation combining depth and RGB information. The 3D voxel representation is created with the same height and width as the original image, and with depth determined by the difference between the maximum and minimum depth values found in the images. We then quantize our depth values into 10 intervals and feed our input into VGG net.

![Model 2 Diagram](image3.png)

- **Model 3: Color Mapping + Parallel VGG Model**. We color-mapped the depth information into an RGB (299x299x3) image using JET color map. In this way, depth information is represented as colors and hence can be picked up by VGG. The regular RGB image is fed into different VGG. We then feed the output of both VGGs into several fully-connected layers to generate an output of estimated $(x, y, z)$ location of the target object.

![Model 3 Diagram](image4.png)

**Results**

- **2D VGG model**: we only apply a single convolutional layer to convert the depth information, so the error is relatively large.
- **3D Voxel VGG model**: we quantize our depth information into 10 intervals and the error mainly comes from the depth discretization.
- **Color Mapping + Parallel VGG model**: this model mostly reserves the depth information though color-mapping, so it performs better than the other two models.

![Attention Map](image5.png)

![Training Strategy](image6.png)

**Conclusion**

- We implemented object localization based on three different models. The experiment results show that model 3 outperforms the rest and achieves 18.67% and 20.15% relative improvement in test error, compared to model 1 and model 2.
- To improve this result, we need to 1) generate more data or 2) add bounding and pixel-wise label to our dataset.