Flip the Switch

FRI Final Presentation

Elin Park, Kevin Ayala, Efren Mendoza, Jay Upadhyaya, Sai Tanuj Madisetty

Problem

BWI bot does not have the ability to detect and interact with light switches.

As such, implementing it would lead to increased potential applications for the robot and the services it could offer.

However, it needs to be able to be done efficiently (in real-time)

- The frame needs to be analyzed relatively quickly
- The algorithm therefore needs to be fast

Background

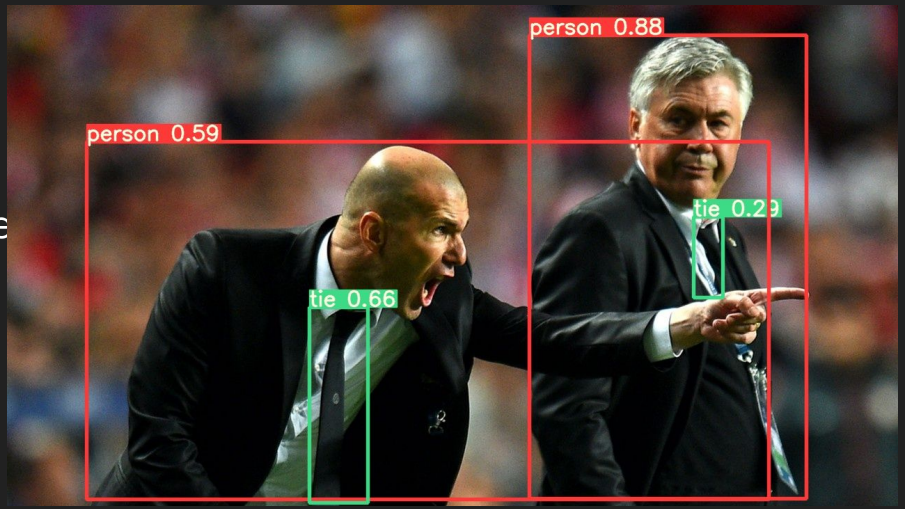
YOLO

State of the art object detection machine learning algorithm

Only passes the image through the neural net once

Outputs a bounding box with a prediction value

Advantages: Speed, Ease, Lightweight



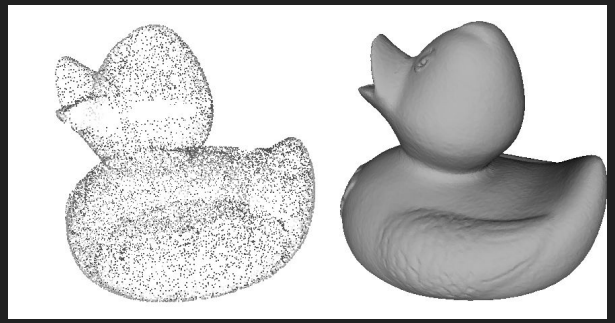
3D RECONSTRUCTION

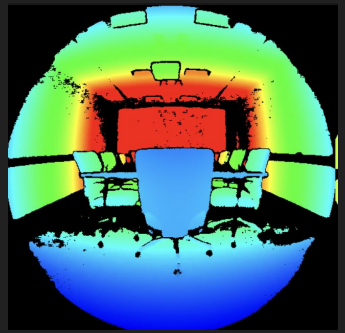
Active

Uses lasers or infrared light

Creates a depth map

From the depth map, creates a point cloud



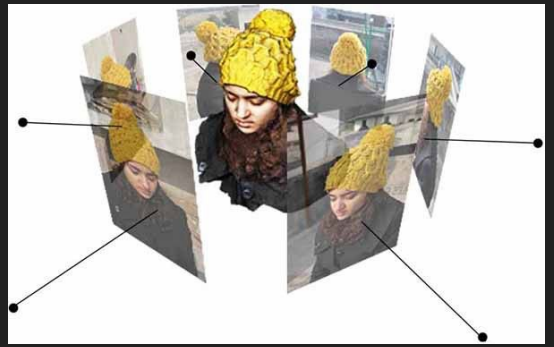


Passive

Uses color images

May take multiple images

Transformations



Approach

Training the Algorithm

1. Compile various pictures of light switches
2. Label images of light switches and create bounding boxes with Makesense.ai
3. Split into training and validation set
4. YOLOv5 - Tested different epoch sizes to increase precision and recall values.

YOLOv5 training

Testing with the Kinect

Use pyk4a to capture a single capture from playback

Color Image

Obtain the color image from the capture

Pass the image in through the YOLOv5 algorithm

Draw the box around the prediction returned by YOLO

Point Cloud

Obtain the depth image from the capture

Transform 2d array of depth image to a depth point cloud array

Connecting to ROS

- Since YOLOv5 is a python algorithm, rospy had to be used instead
- The implementation of rospy is essentially the same as roscpp, just with python scripts rather than c++ source files
- Finally, we published the pointcloud to ROS

EVALUATION

Evaluation

We increased our epoch size from the standard 40 epochs to 120 epochs due to the low amount of training data

Precision value - 1.0

- Correct results (No FPs)

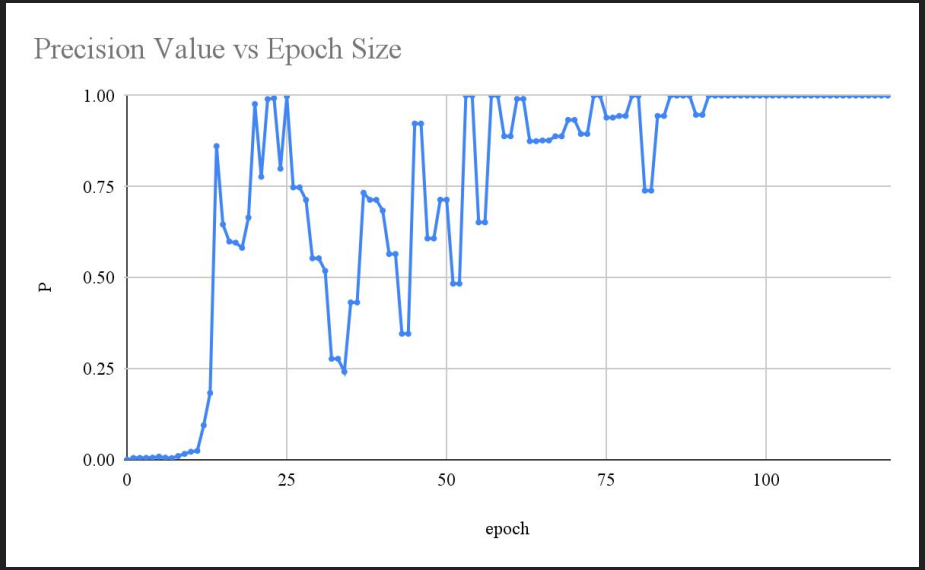
Recall value - 0.9

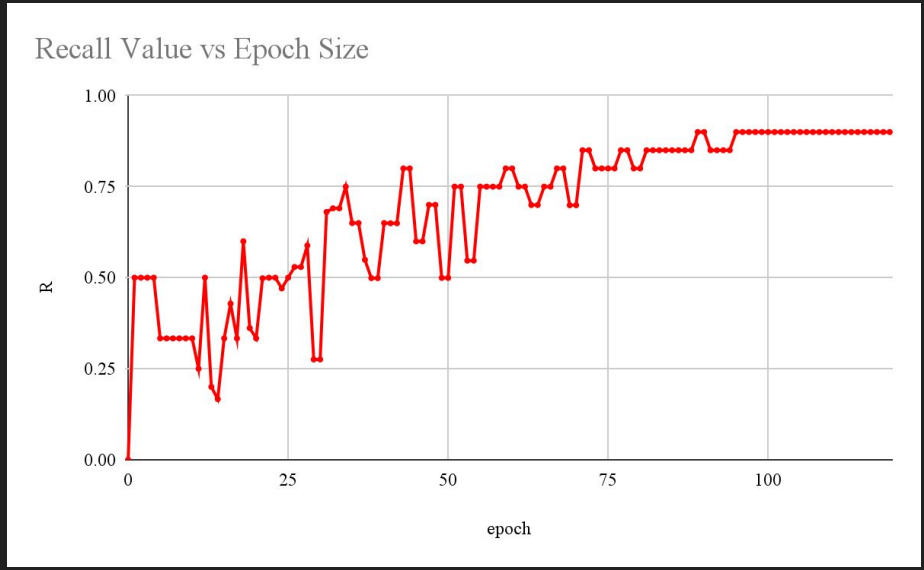
- Successful retrievals (Few FNs)

Improve training settings

- Train at a higher resolution
- Use more data points (more pictures)

Evaluation- Precision and Recall





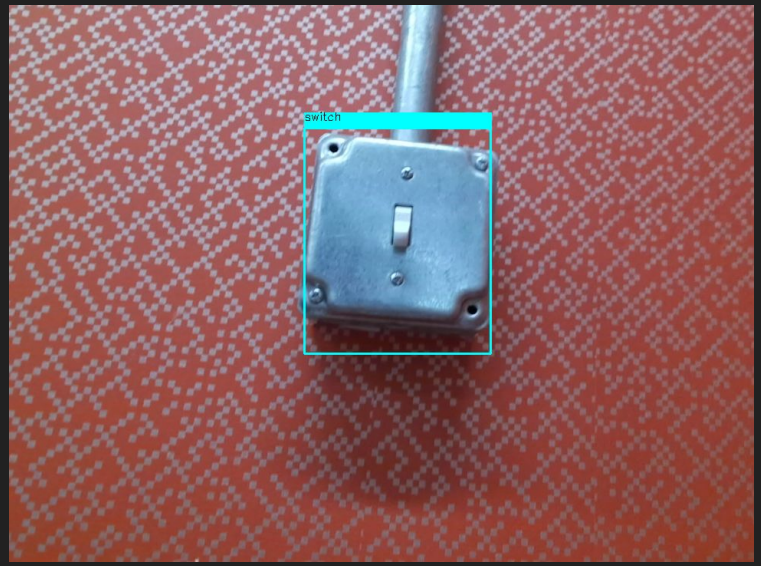
$$\text{Precision} = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

$$\text{Recall} = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

RESULTS

Recordings and Results

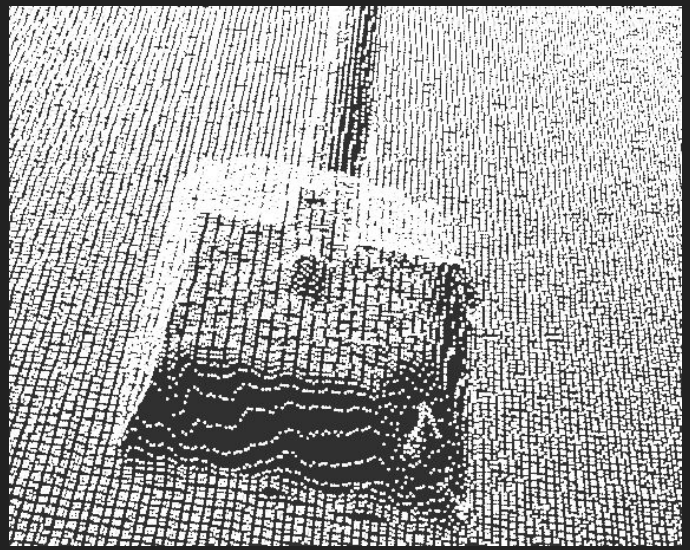
- We had 5 recordings of the light switch from various angles and perspectives
  - Pan up, down, left and right
- Detected light switch even as it is heading out of frame



RESULTS

Able to publish and visualize the cloud points on rviz for all recordings

Next Step: Isolate the light switch and publish only that point cloud to rviz



Conclusion

Problem

- How can we get the BWIbot to recognize everyday objects?

Approach

- Trained YOLOv5 algorithm, processed with pyk4a, and published to rviz

Results

- Detected and marked light switch, published point cloud of the frame

Future Application

- Service robots rely heavily upon object detection

Next Steps

- Move the robot to the switch (spatial tf), use MoveIt to flip it

Questions.