# ACCIO


Youya Xia
Yingxin Wei
Xiaofei Chen
Ziqian Qiu

# Goal of our Project

Command (tell TurtleBot what we want)

TurtleBot go to Baxter

Baxter pick up that object

Baxter hand it to the TurtleBot

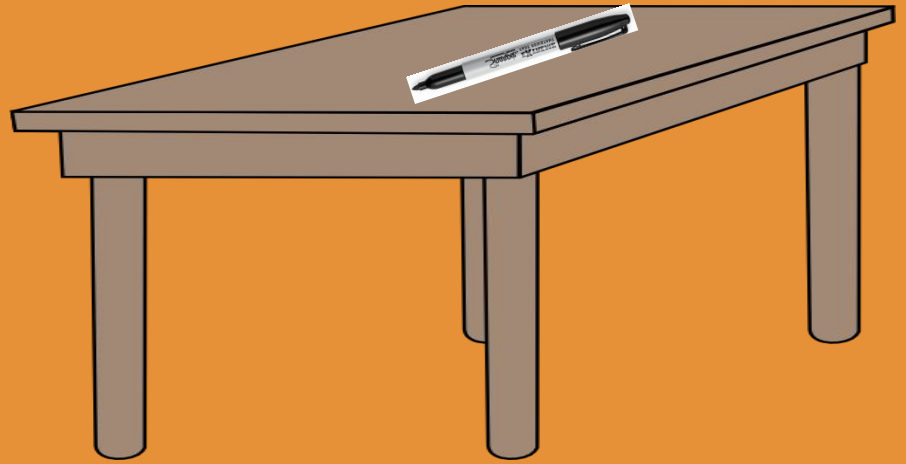
TurtleBot hand the object we want to us

**Please give me the marker!**

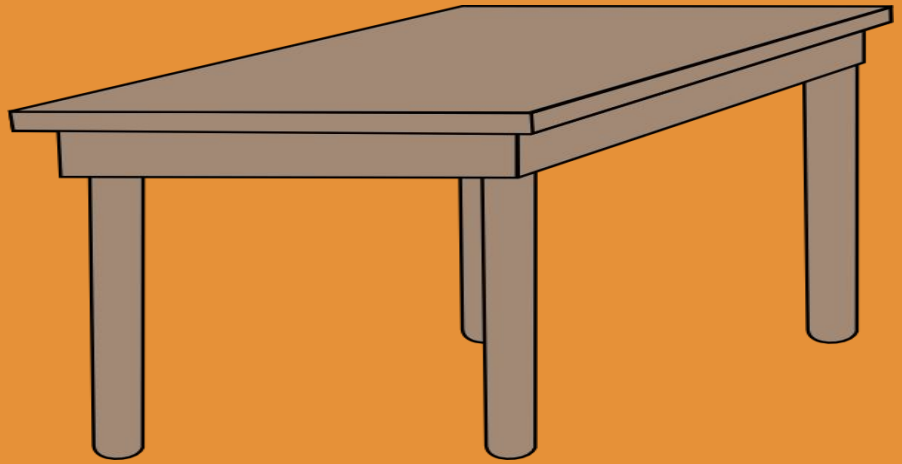

Move

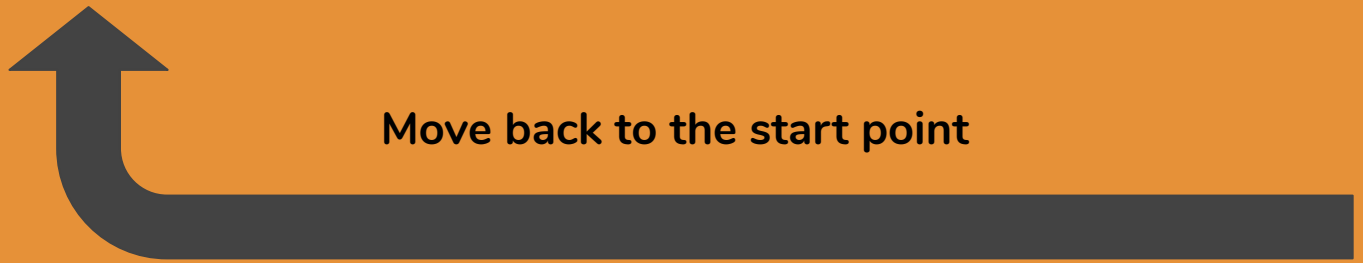

Communication

**Detect the object and pick it up**



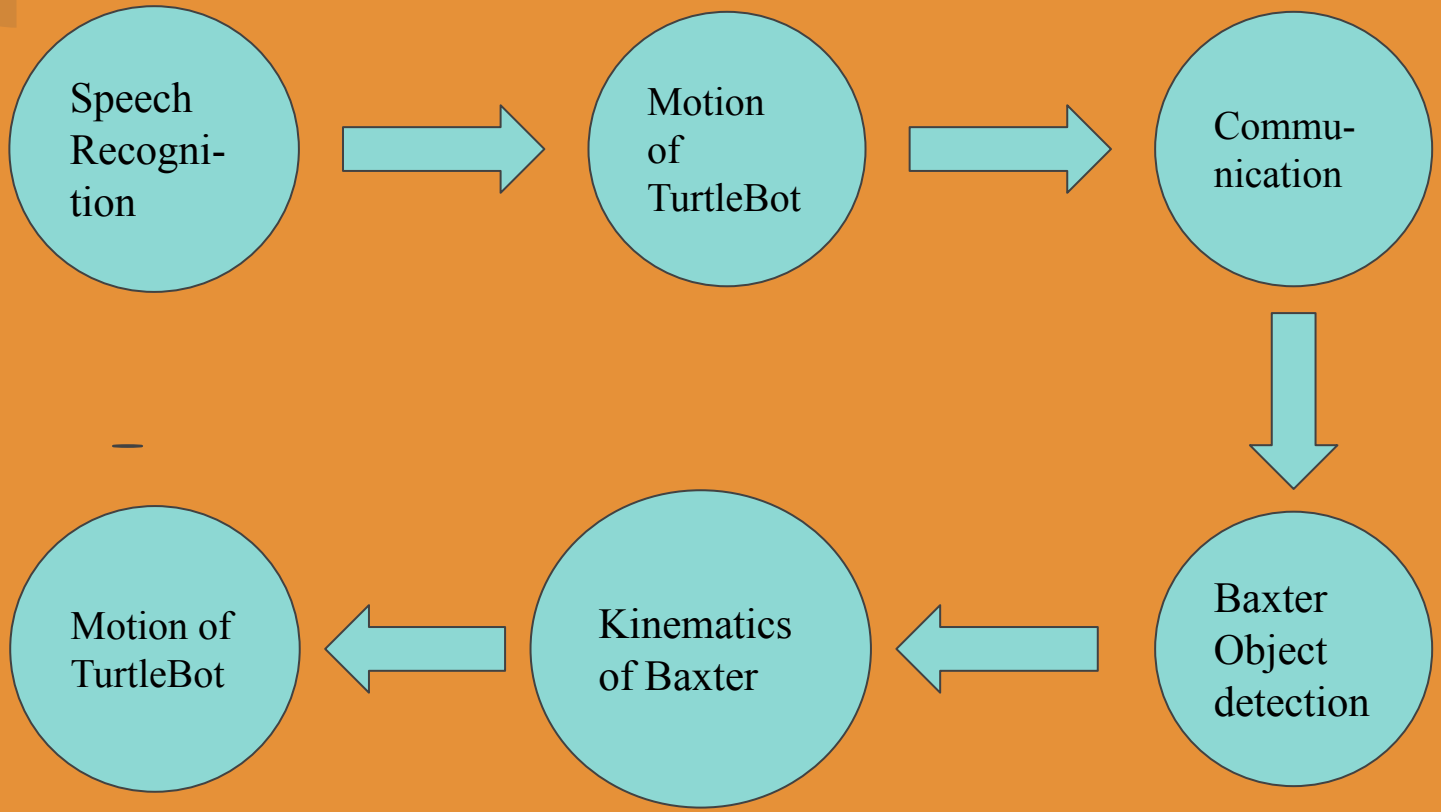

**Put the marker on TurtleBot**



Move back to the start point

# Steps:


Speech Recogni-tion
Motion of TurtleBot
Commu-nication
Baxter Object detection
Kinematics of Baxter
Motion of TurtleBot

# 1.Speech Recognition

semantic segmentation. We will analyze voice commands from our team members, which is the framework of goal for each task. Once the Baxter matches the keyword of the commands with certain classes, it would reach the corresponding item labelled by vision recognition

First, we have finished the speech recognition part and simple key words capture by using speech recognition API provided by Google. Once we say something like “Please take the xx for me”, the program will be able to recognize the whole command and print “Task xx”. For other irrelevant command like “Please take the course CSCI5551 for me”, at the current stage, the program will print “No such task”.

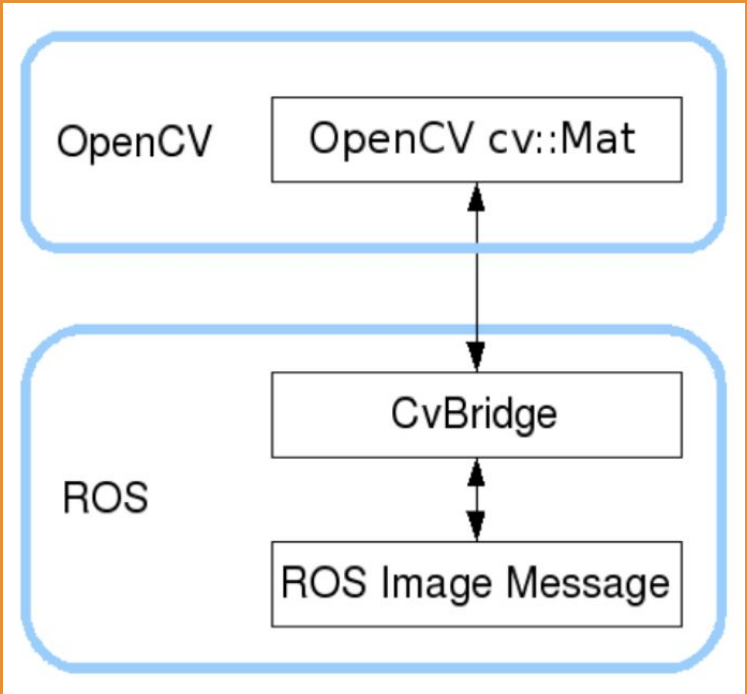
# Image

# 2. Communication

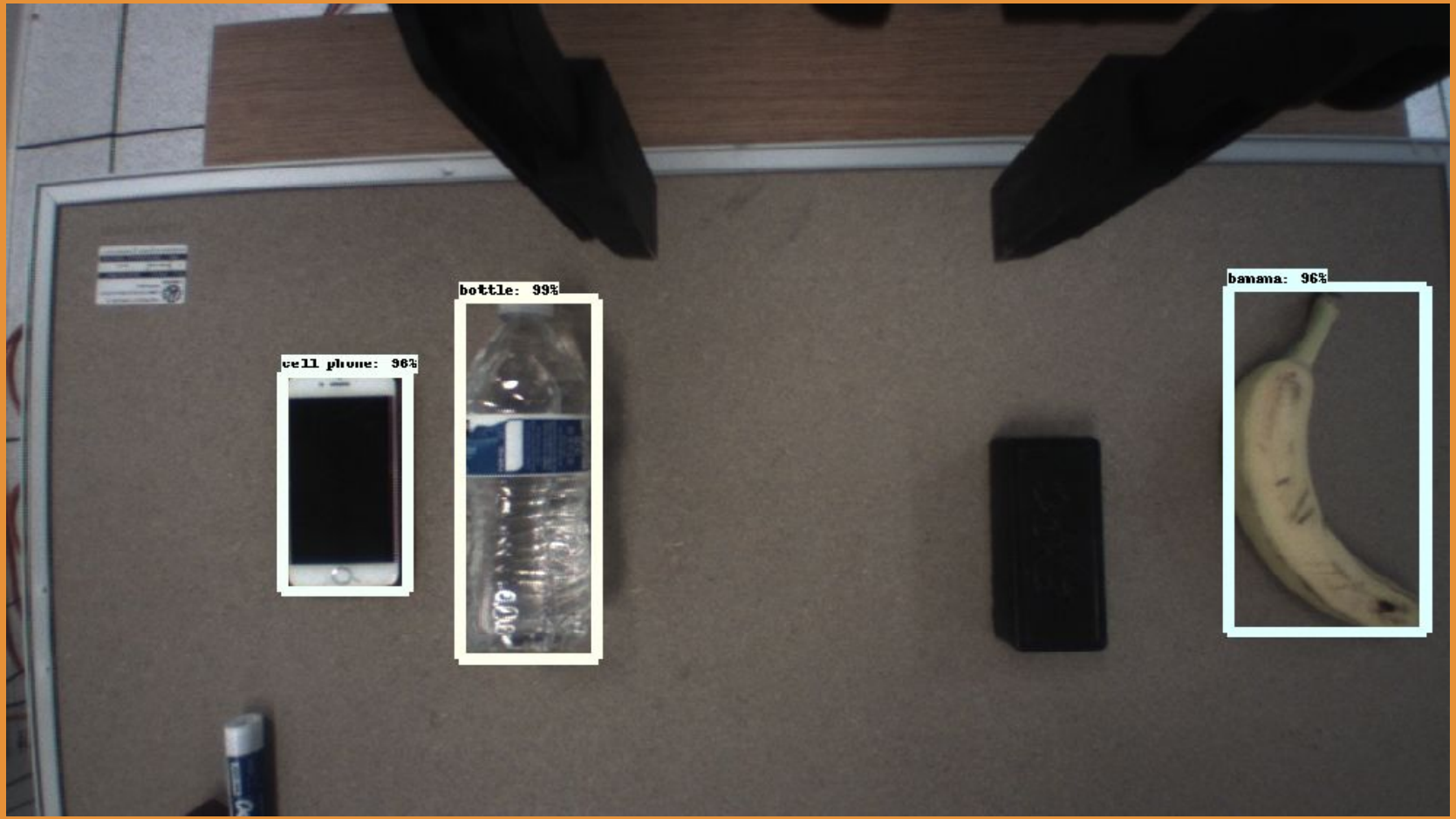
- 915MHz modules, Readytosky 3DR Radio Wireless Telemetry Kit
- Standard TTL UART
- Range of several kilometers
- Every object has a corresponding Alphabet to be sent

# 3. Baxter Object Detection

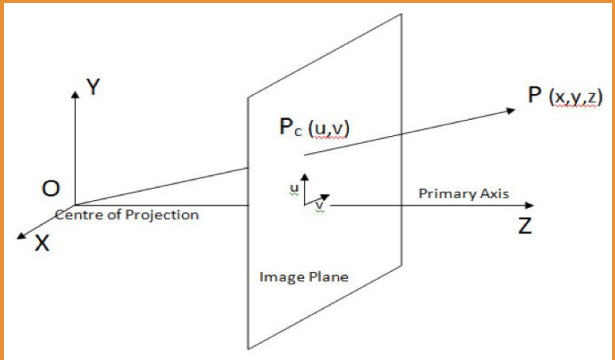
- Use pretrained model (Faster RCNN)
- labeled 2000 images for **six** different objects
- Test our training results on Baxter's image later




- Get image from Baxter’s right hand camera
- Interact between OpenCV and Ros using CvBridge
- Detect the object and enclose it in a frame
- Return the center coordinate of the box


cell phone: 96%
bottle: 99%
banana: 96%

# 4.Baxter Kinematics




**B = (Pp – Cp) * cc * d + Bp + Go**

where:

- B = Baxter coordinates
- Pp = pixel coordinates
- Cp = centre pixel coordinates
- Bp = Baxter pose
- Go = gripper offset
- cc = camera calibration factor
- d = distance from table

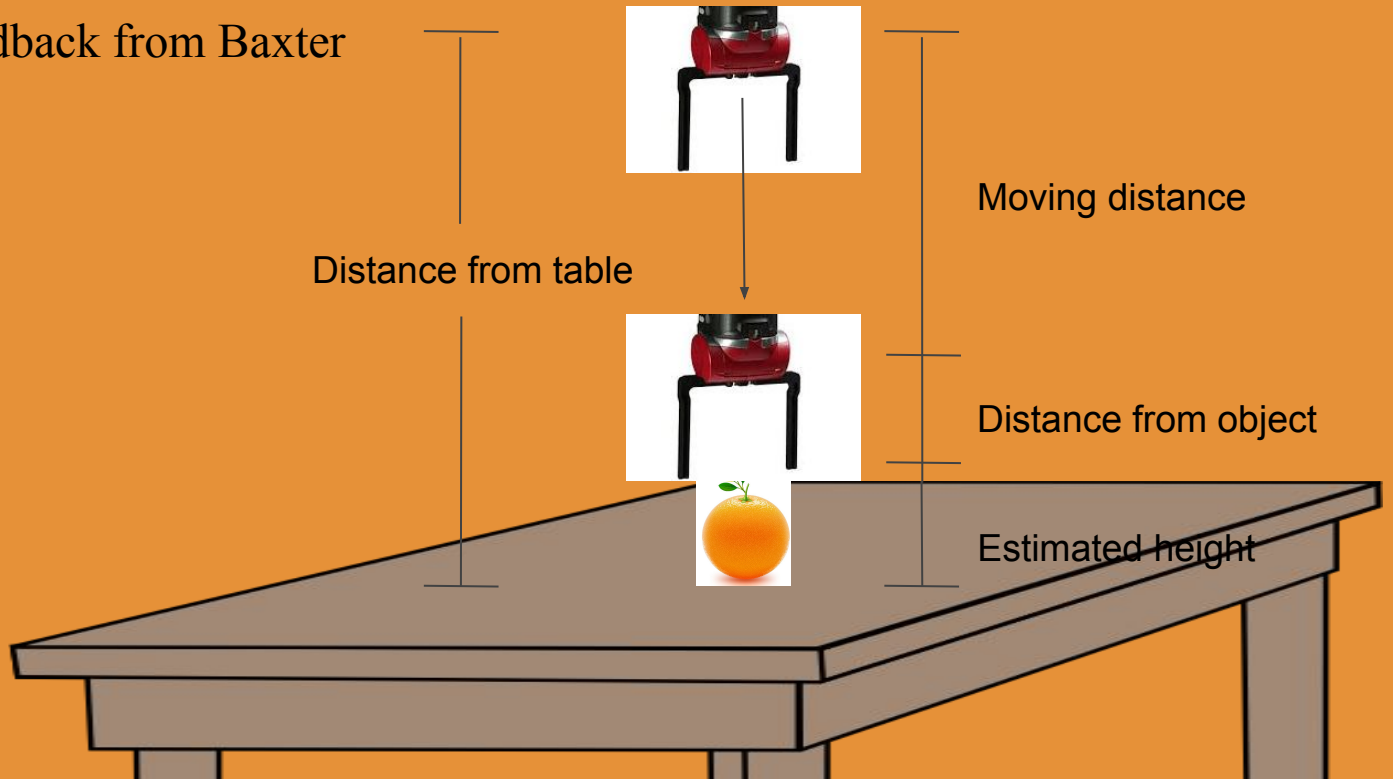
- Image pixel to workspace coordinate conversion
- Right arm point vertically down to the table
- Camera position is fixed at a point
- Detect the distance from the table using IR range sensor
- Camera resolution 960x600
- Inverse Kinematics-MoveIt!

Video : https://youtu.be/DPxIGrnUNpo

# 5.TurtleBot Motion

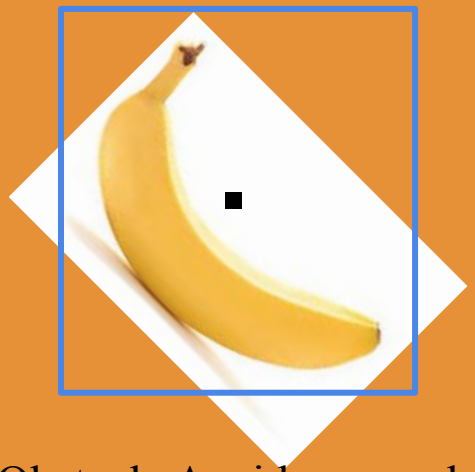
# Challenges & Possible Solutions

1) Lack feedback from Baxter



# Challenges & Possible Solutions

2) Haven’t extract the orientation of the objects during detection.



3) Obstacle Avoidance and autonomous navigation for Turtlebot

# Future Work

We are almost done.  

# References


https://www.robot-advance.com/EN/art-turtlebot-2-complete-assembly-1189.htm

http://sdk.rethinkrobotics.com/wiki/Home

http://sdk.rethinkrobotics.com/wiki/Worked_Example_Visual_Servoing

https://gifer.com/en/bf6

https://www.amazon.com/Readytosky-Telemetry-Standard-Version-controller/dp/B01DHV4DVA

# Questions?