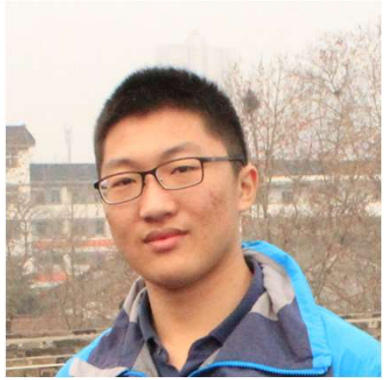
Team



Lihao Zhu

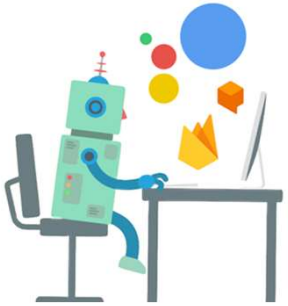
Yuanhai Tan

Xuewei Wang

Jiayun Zhong

Dōkō

General Usage Robot



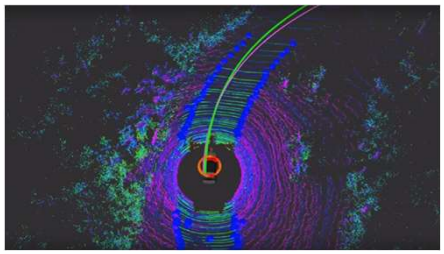
General
Utility
Neuro-network
Driven
Advanced
Mobile-Robot

In collaboration with

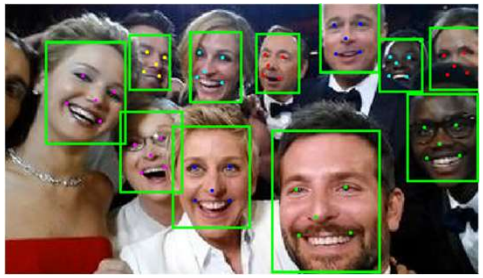
OHMNILABS

Overview: We want to enable a robot to do...

- **Navigation** - move from one place to another place automatically.



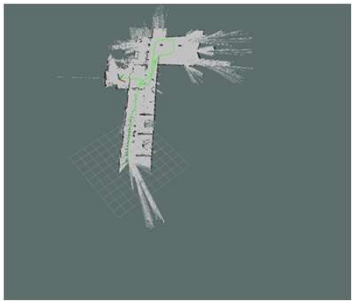
- **Face Recognition** - recognize up to two hundred faces.



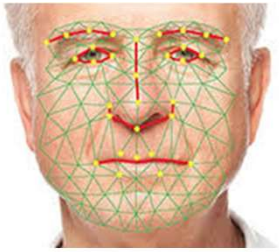
- **Voice Control** - cloud computing powered voice assistant.

Overview: What we have done

- **Navigation** - System completed, fine tuning some parameters.



- **Face Recognition** - Model finialized, integrating with other systems.

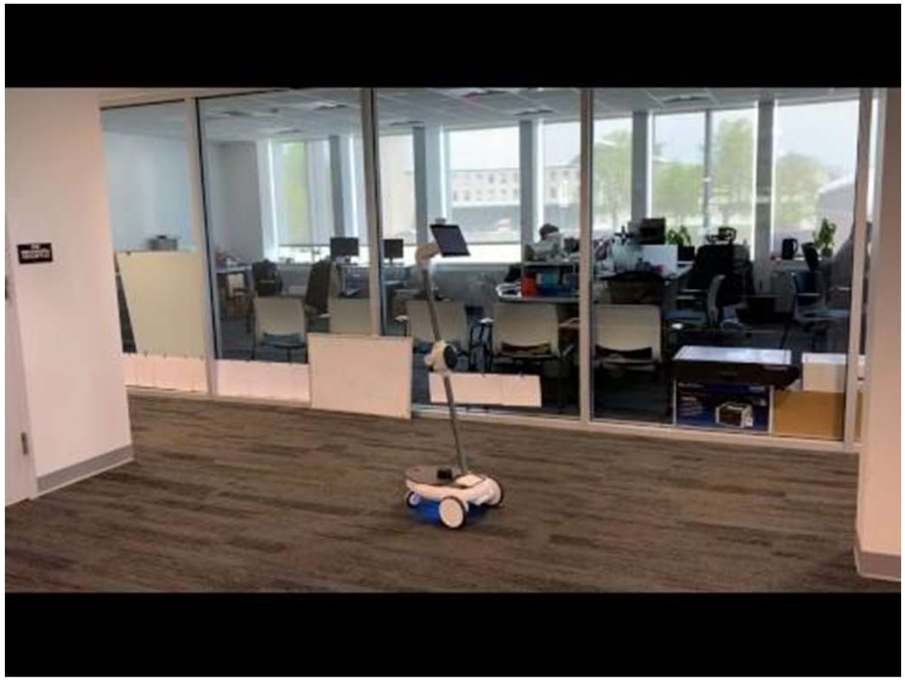


- **Voice Control** - Drive commands and simple conversion.

Navigation: I hate glass walls

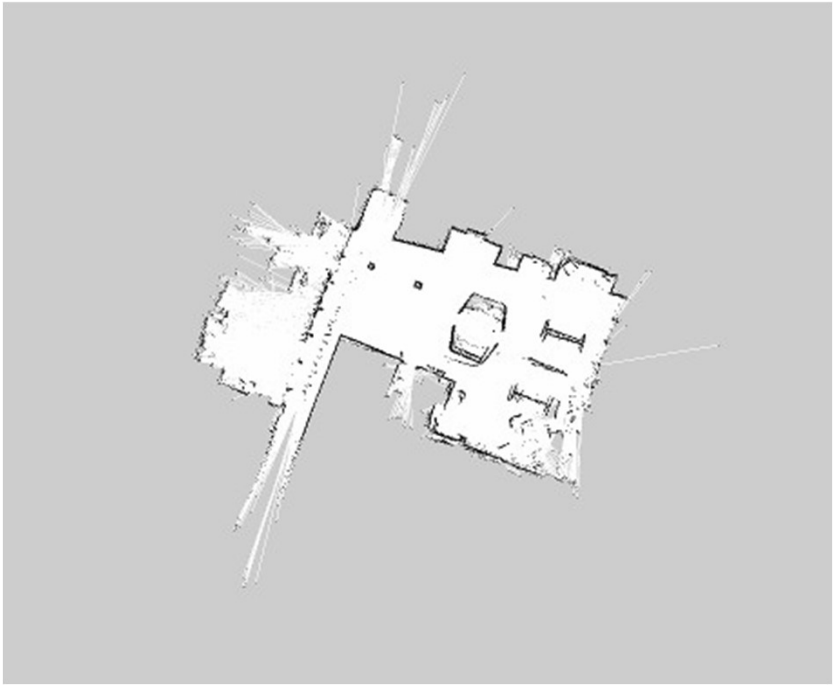
- Fully automatic self driving system
- A lot of code

- Fusion of different sensors



Navigation: which can’t be seen.

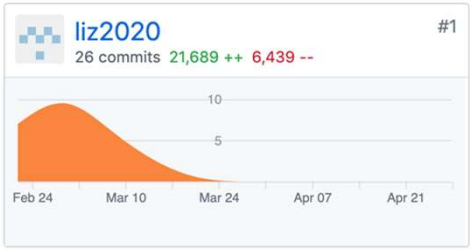
- We introduced a drastic change in the map
- People moving around the robot
- Narrow aisle



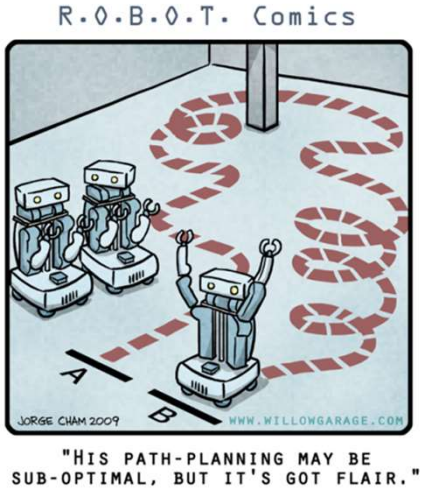
Timeline

Jan 21 ~ Feb 24
Communicate with the OhmniLabs

Feb 24 ~ Mar 24
Training Model



Mar 24 ~ Current
Working on Navigation

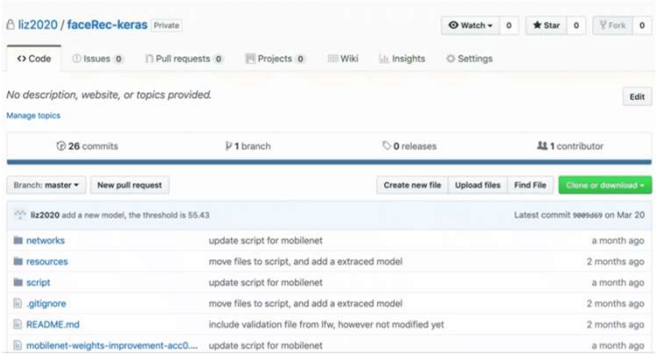


http://clipart-library.com/communication-clipart.html
http://wiki.ros.org/navigation?action=AttachFile&do=get&target=nav_comic.png

Face Recognition: Model Training

Implement code that

- **Train** model according to different networks
- **Save** models and restart training from checkpoints
- **visualize** training process using tensorboard
- **Test** on LFW (labeled faces in the wild)
- **Deploy** the model to android or web



Provide documentation for setting up environment, training model, and troubleshooting.

ANACONDA.

Keras

Dataset: CASIA-WebFace

Face Recognition: Training Output

Train model that

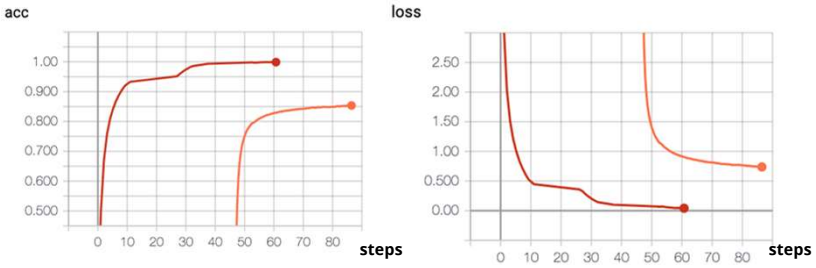
- **92.06 %** accuracy on LFW using **Mobilenet**
- **82.283 %** on LFW using **simplified LFR**
- Use Mobilenet so Xuewei can work on the deployment early

Both could obtain > 99% on LFW in theory, but

- Takes time to fine tune the parameters.
- Simplified the LFR due to time restriction.
- The pre-trained model and training code for LFR is not released.

VGG-Face:	553 MB
Facenet:	93 MB
Mobilenet:	**22 MB**
LFR:	? MB

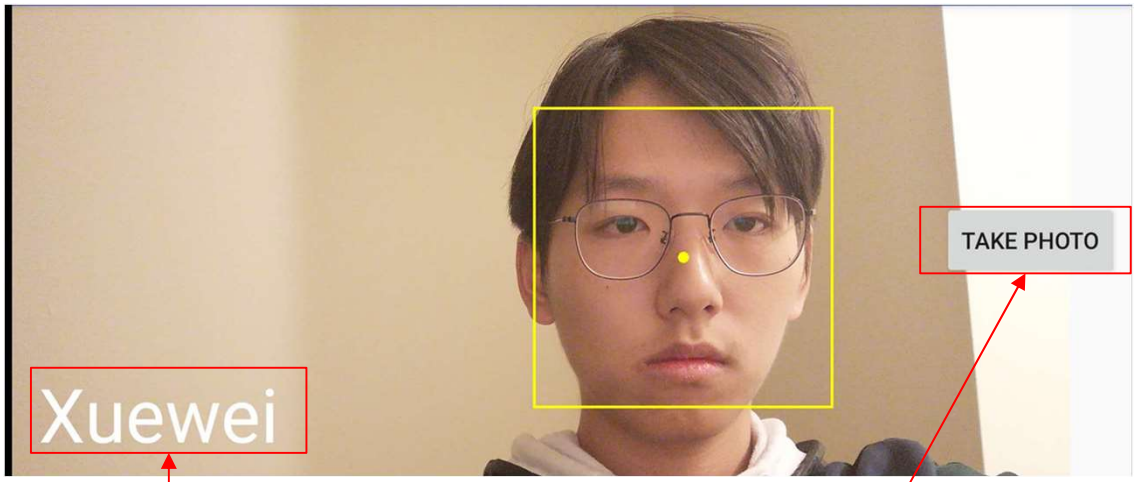
Smaller size
= less parameters
= shorter computation time



Lihao Zhu

Android App: User Interface Design

- Camera Activity: real time face detection and recognition.



Name of the current user, whoever's face occupies the greatest area.

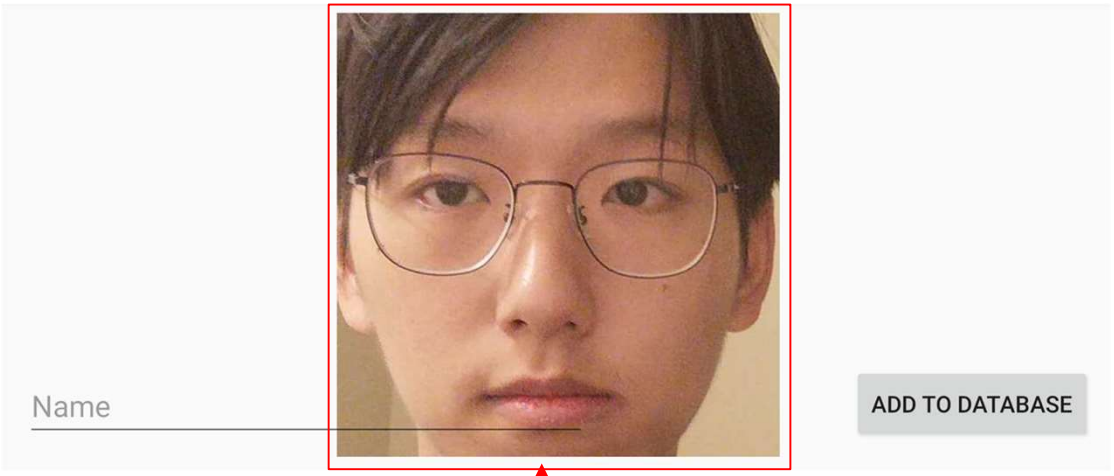
"TAKE PHOTO": initiate the face registration activity with the face detected.

Android App: User Interface Design

- Communication Activity: doko interact with the user by starting a conversation.

Android App: User Interface Design

- Face Registration Activity: where users add their faces to the database.



The user's face detected by face detection model is displayed here.

The Android App: Technical Details

- Video Processing: Camera API to perform face recognition on each frame.

- Face Recognition: two models integrated by OpenCV and TensorFlow Lite. A SQLite database storing face feature vectors along with names.

- Communication:
  - recognizing speech (CMU Sphinx)
  - generating a response (DialogFlow)
  - response to speech (TextToSpeech)

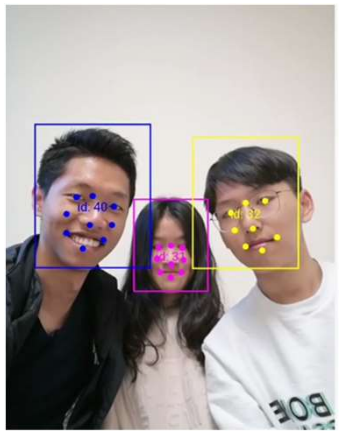
Timeline and Individual Progress

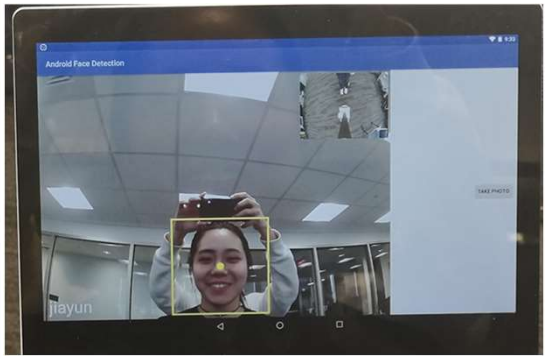
Jan 21 ~ Feb 24
Early Experiment on Keras

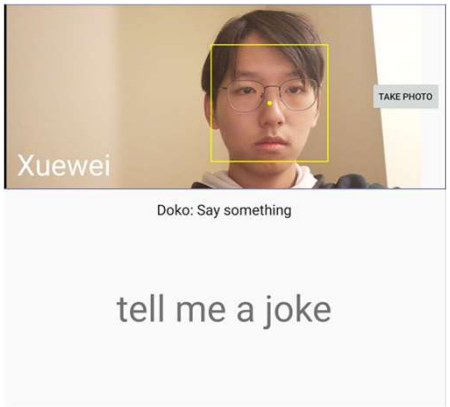
Feb 24 ~ Mar 24
First Prototype: Face Detection

Mar 24 ~ April 10
Second Prototype: Face Detection + Face Recognition

April 10 ~ May 1
Intermin Prototype

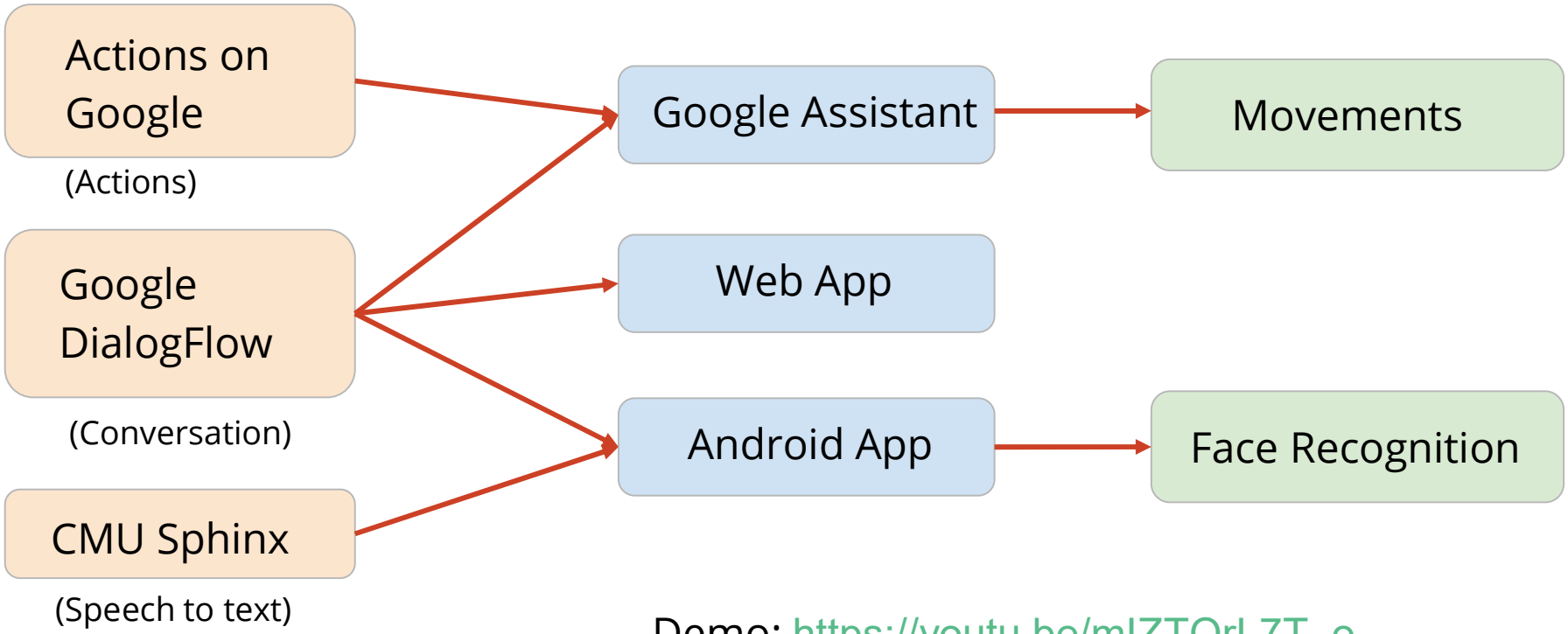






- 18 commits, ~2000 lines of Java

Voice Assistant



Demo: https://youtu.be/mIZTQrL7T_o

Summary and Plan

Summary:

Navigation: She moves on herself.

Face Recognition: Train model with different structures and deploy it.

Voice Assistant: User command movements, basic conversations, incorporate with face recognition.

General Plan for next semester:

- Make 3 parts work together smoothly and test in different scenarios.