# Use Case

- 53 percent of people currently use smart home devices
  - Increasingly trending towards more IoT devices
  - Deaf and Mute people are currently unable to use these devices
- Our project aims increase accessibility by creating a smart home device powered only by gestures
- Proof of concept for lightweight add-ons for existing technology that lack accessibility features
- ECE Areas
  - Software Systems
  - Signals

User Interface
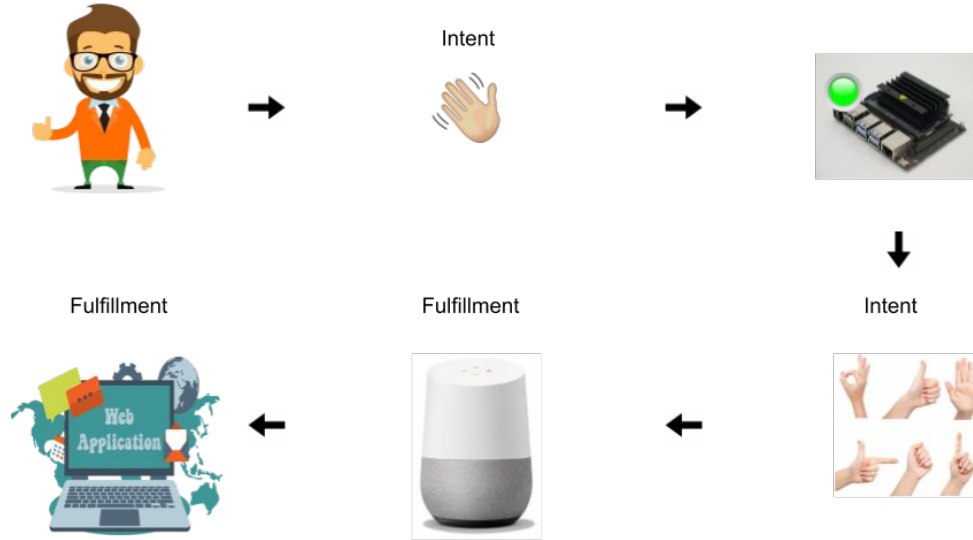
Key Technical Challenges

- Real Time Gesture Recognition Algorithm
  - Recognizes gestures with speed and accuracy
  - Ability to put gestures together into a coherent word or command
- Integration with Google Assistant
- UX/UI
  - Easy to use and accessible for deaf and mute users
  - Similar capabilities to standard voice
  - Able to communicate back information in a non-verbal manner
  - Minimize privacy concerns

Requirements

- Speed:
  - Algorithm recognizes commands in 50 milliseconds
  - After input received, respond with information within 2 seconds after given gesture
- Accuracy:
  - Successfully recognizes commands 80 percent of the time
- Data Reliability:
  - Communicates the query to Google Assistant 99.9 percent of the time
  - Respond with information about query result 99.9 percent of time
- Acceptable Conditions:
  - Works in indoor lighting (500 to 1000 lux)
  - Works within 5 meters of camera

Common Commands

- Ok Google (activation)
- What's the weather?
- Set/cancel an alarm at _____ time?
- Set/cancel timer for ___ minutes
- Play/stop music
- How's the traffic to work?
- Stop
- Fingerspelling for custom inputs
    - Text
    - Numbers
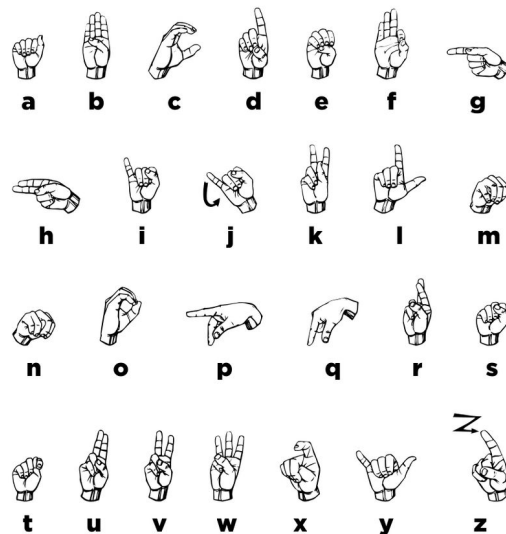
Solution Approach: Signals

- **Feature Extraction**
  - Generate a skeleton from an image
  - Locate all the joints of the hand
- **Gesture Recognition from Skeleton**
  - From the joint coordinates identify gestures or fingerspelling letters
  - Skeleton approach uses much less data than using raw images
- **Potential Algorithms**
  - Neural Networks
  - Support Vector Machine
  - K Nearest Neighbors
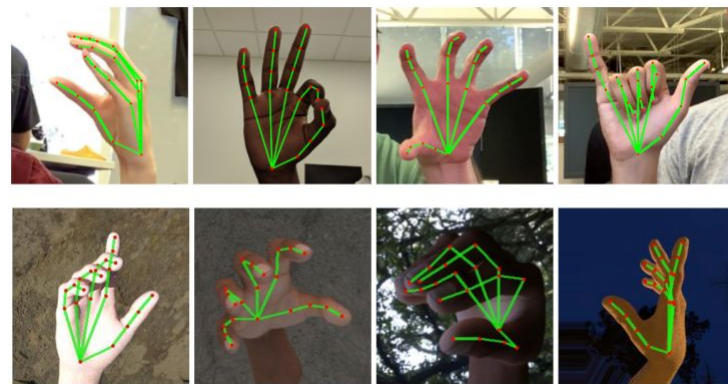
Solution Approach: Software

- Google Assistant has one of the best natural language processing
  - Identify (Intent) -> Execute (Fulfillment)
- Skip the parsing of intent to directly feed signed/gestured text to Google Assistant
- Jetson will interface with a web application that will display Google Assistant responses
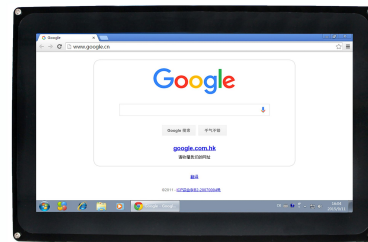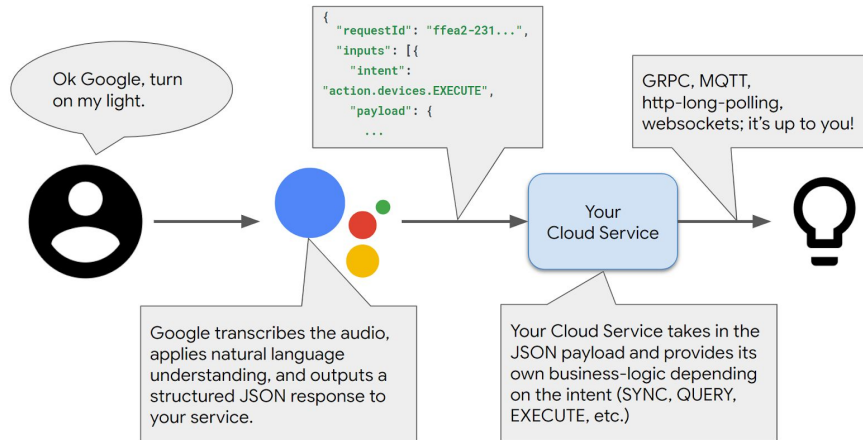
Solution Approach: Hardware

- Nvidia Jetson Nano
    - 70 x 45 mm
    - GPU: 128-core Maxwell
    - CPU: Quad-core ARM A57 @ 1.43 GHz
    - NVIDIA JetPack SDK
- Google Assistant on Jetson eliminates need for additional Google Home hardware
- RGB-D Camera
- Speaker
- Display
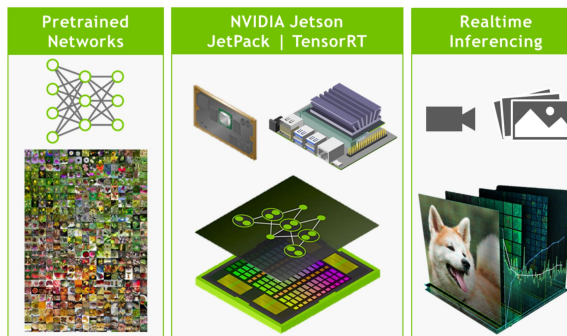
Testing, Metrics, and Validation

- When running gesture recognition model on Jetson Nano, track program time of completion
- Generate large video dataset of gestures in range from minimum to maximum acceptable distance and minimum to maximum acceptable lighting conditions from various different people
- Use above data to initially test accuracy and speed transitioning later to live gesturing
- Store log of errors to ensure, errors were not caused by failure to communicate with Google Assistant

Testing, Metrics, and Validation (use case)

- We targeted all our metrics to match a Google Home's performance in speed and accuracy
- Google Home already commercially successful
- We will try to also do user testing to find real live feedback from deaf and mute people

Division of Labor

Sung	Claire	Jeff
● Skeletal tracking of hand ● Gesture recognition Implementation and Optimization ● Learn Fingerspelling	● Hardware selection and setup ● Google Assistant SDK on embedded ● Communicating with the Webapp and displaying responses ● Circuits ● Learn fingerspelling	● Web App to interface with the Nvidia Jetson ● UI/UX ● Images and Features ● Physical Enclosure ● Learn fingerspelling

Schedule

TASK TITLE	TASK OWNER	START DATE	DUE DATE	DURATION
Collection				
Acquire and set up Jetson with microUSB	CY	2/3/20	2/7/20	4
Research and buy RGB-D Camera, speaker	CY	2/3/20	2/5/20	2
Research and buy screen and LED	CY	2/5/20	2/7/20	2
Revisions based on project proposal	CY, SH, JH	2/5/20	2/7/20	2
Find and request datasets	SH, JH	2/10/2020	2/12/20	2
Collect and sort through datasets	SH, JH	2/12/2020	2/17/20	5
Implementation				
Set up Google Assistant queries on Jetson	CY	2/12/2020	2/17/20	5
Wire LED and RGB-D Camera	CY	2/13/2020	2/17/20	4
Wire speakers and screen	CY	2/14/2020	2/18/20	4
Detecting hand from background	SH	2/5/2020	2/11/2020	6
Skeletal Extraction with Clear Background (Static)	SH	2/12/2020	2/18/2020	6
Skeletal Extraction (Static)	SH	2/19/2020	02/26/2020	7
Skeletal Tracking (Dynamic)	SH	2/27/2020	03/04/2020	7
Skeleton Gesture Recognition (Dynamic)	JH,SH	3/23/2020	03/30/2020	7
Image normalization	JH,sH	3/17/2020	03/22/2020	5
Slack for Skeletal Recognition	SH	3/31/2020	04/05/2020	5
Skeletal Recognition Optimization	SH	4/6/2020	04/13/2020	7
Web Application (Design and Research)	JH	2/6/2020	2/9/2020	4
Web Application (Create Initial Functionality)	JH	2/10/2020	2/16/2020	7
Web Application (Generate Custom Response)	JH	2/17/2020	2/23/2020	7
Web Application (Hosting and AWS)	JH	2/24/2020	03/01/2020	7
Web Application (Visual)	JH	03/31/2020	04/05/2020	6
WebApp with Jetson & Google Assistant	JH,CY	3/3/2020	03/06/2020	3
Enclosure	JH	04/06/2020	04/12/2020	6
Integration between pre-trained networks and Jetson	CY,SH,JH	3/17/2020	3/20/20	3
Slack				
Testing and Optimization				
Generating testing dataset	CY, SH, JH	2/24/2020	3/6/20	12
Python script to generate large random tests on Jetson	CY	3/3/2020	3/6/20	3
Recognizing hand + skeletal on Jetson	CY, SH, JH	3/20/2020	3/23/20	3
Recognizing a still gesture (command and fingerspelling)	CY, SH, JH	3/26/2020	3/31/20	5
Recognizing a series of gestures with fingerspelling	CY, SH, JH	3/26/2020	3/31/20	5
Tweaking Jetson gesture understanding	CY	3/31/2020	4/2/20	2
Real-time user testing and tweaking	CY, SH, JH	4/3/2020	4/8/20	5
Final data compilation	JH	4/7/2020	4/11/20	4
Course Logistics				
Design Presentation	CY, SH, JH	2/10/2020	2/16/20	
Design Report	CY, SH, JH	2/25/2020	3/2/20	
Public Demo and Final Presentation	CY, SH, JH	4/23/2020	4/26/20	
Final Report	CY, SH, JH	4/27/2020	5/3/20	

Week columns: WEEK 02/03/20, WEEK 02/10/20, WEEK 02/17/20, WEEK 02/24/20, WEEK 03/02/20, WEEK 03/09/20, WEEK 03/16/20, WEEK 03/23/20, WEEK 03/30/20, WEEK 04/06/20, WEEK 04/13/20, WEEK 04/20/20, WEEK 04/27/20 (each with days M T W R F)