xWalk Design Review

D2: Yasaswini, Jeanette, Ricky

Introduction

- Application Area: Signals and Software
- Problem Statement:
  - Many intersections lack features to facilitate crossing for the visually impaired, some even without crossing signals. Individuals who are training to recognize traffic flow and crossing cues can put themselves at risk.

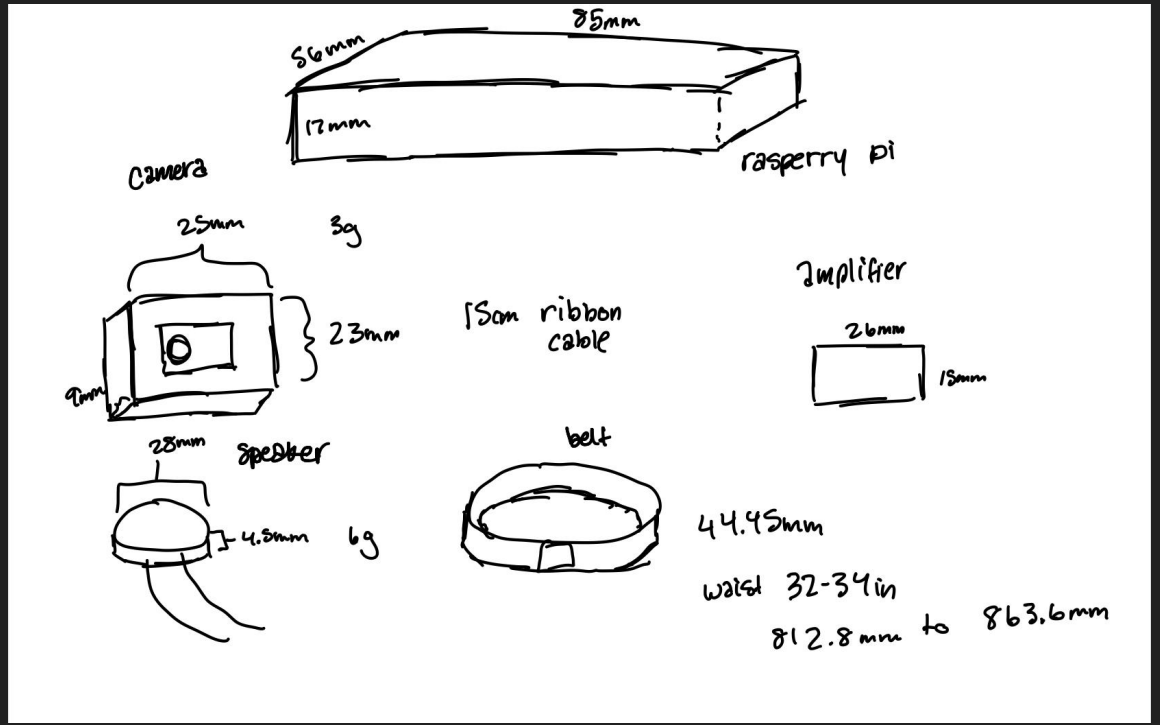
Solution

- New Understanding
  - Past misunderstanding: blind individuals *always* have difficulty knowing when to cross
  - New understanding: difficult to navigate during training period, where user *needs* guidance to make sure they are making the correct timing decisions

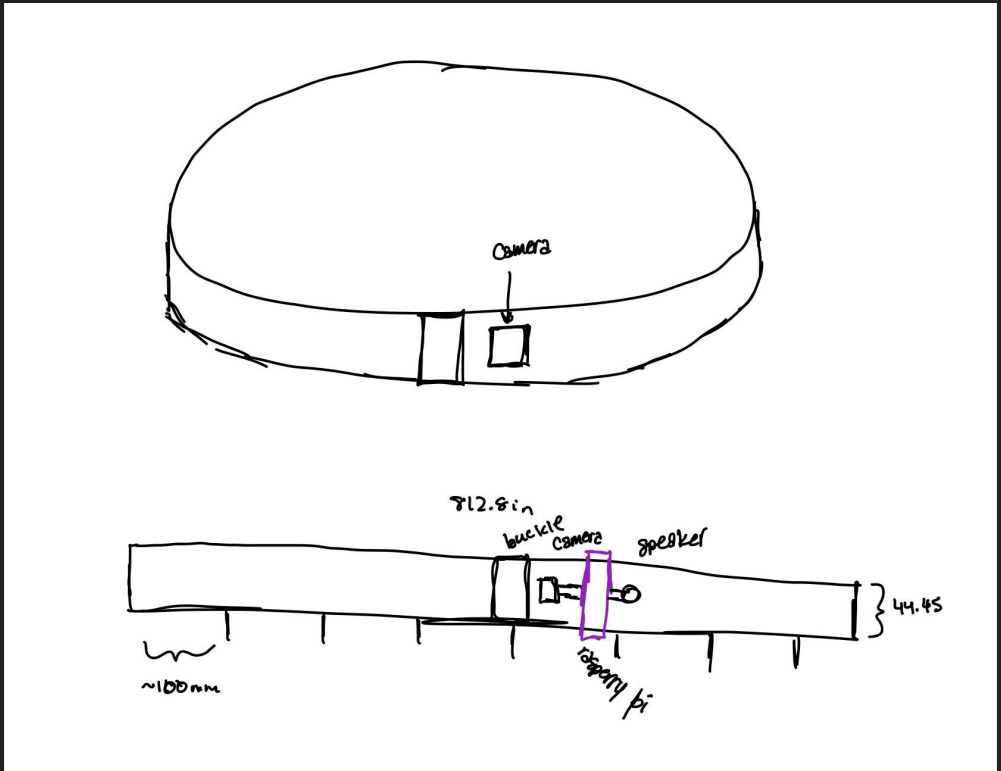
- New Solution
  - Not for consistent use, rather for as-needed use in training and/or for when there isn’t enough audible vehicular traffic

Implementation

- Parts being ordered
  - Raspberry Pi 4
    - Raspberry Pi Camera V2
    - Raspberry Pi Portable battery
  - Adafruit Mini Speaker
  - Nylon belt
  - Enable Button
  - Amplifier



Implementation- Hardware Diagram



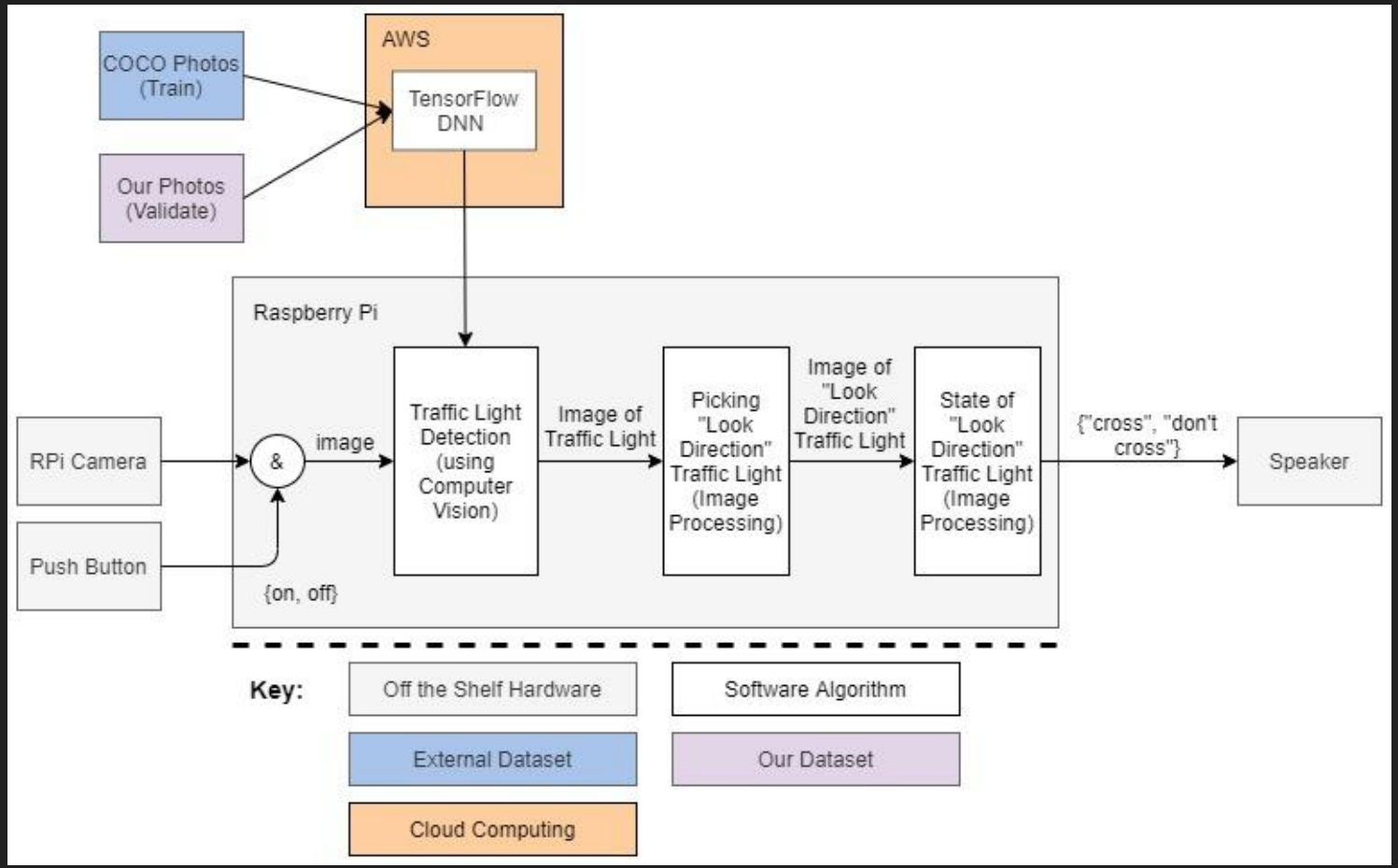
Implementation - Software

- Traffic Light Detection
  - Train using TensorFlow and COCO dataset with tagged traffic lights
    - Train using AWS
  - Validate using our hand-tagged dataset
  - Test on unseen photos using OpenCV
    - dnn class using trained TensorFlow model
    - Input new image
    - Outputs the coordinates of the detected traffic light(s), if any

Implementation- Software

- Look Direction Light
  - Some image processing to deduce which light(s) is most directly facing user
  - Search for combination of circles
  - Calculate distance to the stoplight using the number of pixels
- Light state
  - Image processing on light color (i.e. using rgb filters)

Placeholder slide for block diagram



Metrics and Validation

Metric	Validation Procedure
Latency || < 0.5 sec	- Take video of approaching intersection and count how many frames it takes after light change for system to detect said change - Latency (s) = # frames / (frames/second)
Look Direction Traffic Light Detection Accuracy || > 90%	- As in video setup above, count how many frames in which traffic light is mislabelled - Accuracy (%) = mislabelled frames / total frames
Traffic Light State || False Positive Rate < 2%	- As in setup above, count red lights mislabeled as green when look direction traffic light correctly identified - FPR = frames where red mislabeled as green / total frames
Battery Life || >= 9 hours	- Measure current when system is on - Life (hr) = Battery pack life (mAh) / current (mA)

Risks & Mitigation

Risk	Mitigation
Producing the wrong output for the user to walk when they should not	Using multiple stop lights in the photo for validation; improving accuracy through training
Orientation problems	Notifying if there is no stoplight in the frame
Misclassifying other signals as traffic lights	Lumped into “Look Direction Traffic Light Detection” validation testing - have gathered pictures with both traffic lights and crossing signals with which to test
Impeding objects/No traffic light	Notify the user stoplight is not visible (less risk)
Transitioning states	Incorporate the yellow light status in order to ensure they start walking at the same time consistently

Project Management

Task	Jeanette	Shayan	Yasaswini
Data collection: Take pictures of Morewood/Ellsworth and Amberson/Ellsworth		✓	✓
CV/algorithm training and development			✓
State machine algorithm		✓	
V/V for CV and combined CV + State Machine	✓	✓	✓
Equipment, component procurement	✓		
Assembly - Integration Testing	✓	✓	✓
Assembly - Hardware (Intel Real Time Camera, Audio, Processor)	✓		
Assembly - Software build		✓	✓
Meeting with visually impaired stakeholders	✓		

Schedule Breakdown

TASK TITLE	TASK OWNER	3/1 - 3/8	3/8-3/12	3/15-3/19	3/22-3/26	3/29-4/2	4/5-4/9	4/12-4/16	4/19-4/23	4/26-4/30	5/3-5/7
Design and Implementation											
Proof of concept and hardware prep											
Order critical components	Everyone										
Take pictures at designated stoplights	Shayan										
Organize data set to use in code	Yasaswini										
Research opencv code examples	Yasaswini										
Refresh how to code on raspberry pi	Jeanette										
Design headband to hold raspberry pi and camera and speaker	Jeanette										
Basic Integration											
Connect speaker to raspberrypi	Jeanette										
Connect camera to raspberrypi	Jeanette										
Code for look direction stoplight recognition	Shayan										
Test look direction algorithm	Yas										
Compare realsense camera and rspi camera	Jeanette										
Connect powerbank to raspberrypi	Jeanette										
Code for the state of the stoplight	Shayan										
Test state of the stoplight algorithm	Yas										
Bench test	Everyone										
Sew camera and raspberry pi into headband	Jeanette										
Full Implementation											
Connect raspberry pi and algorithm	Jeanette										
Code auditory feedback	Shayan										
Recognize stoplights	Yasaswini										
Give correct auditory feedback for stoplights	Shayan										
Integration test at stoplight	Everyone										
Slack	Everyone										
Performance Testing and Integration											
Test battery life	Yasaswini										
Test durability and comfort (running, weather, etc)	Shayan										
Tweak parameters (speaker volume)	Jeanette										
Final Report											
Record Video	Everyone										
Final Presentation	Everyone										
Edit Video	Everyone										
Project Due (May 10)	Everyone										