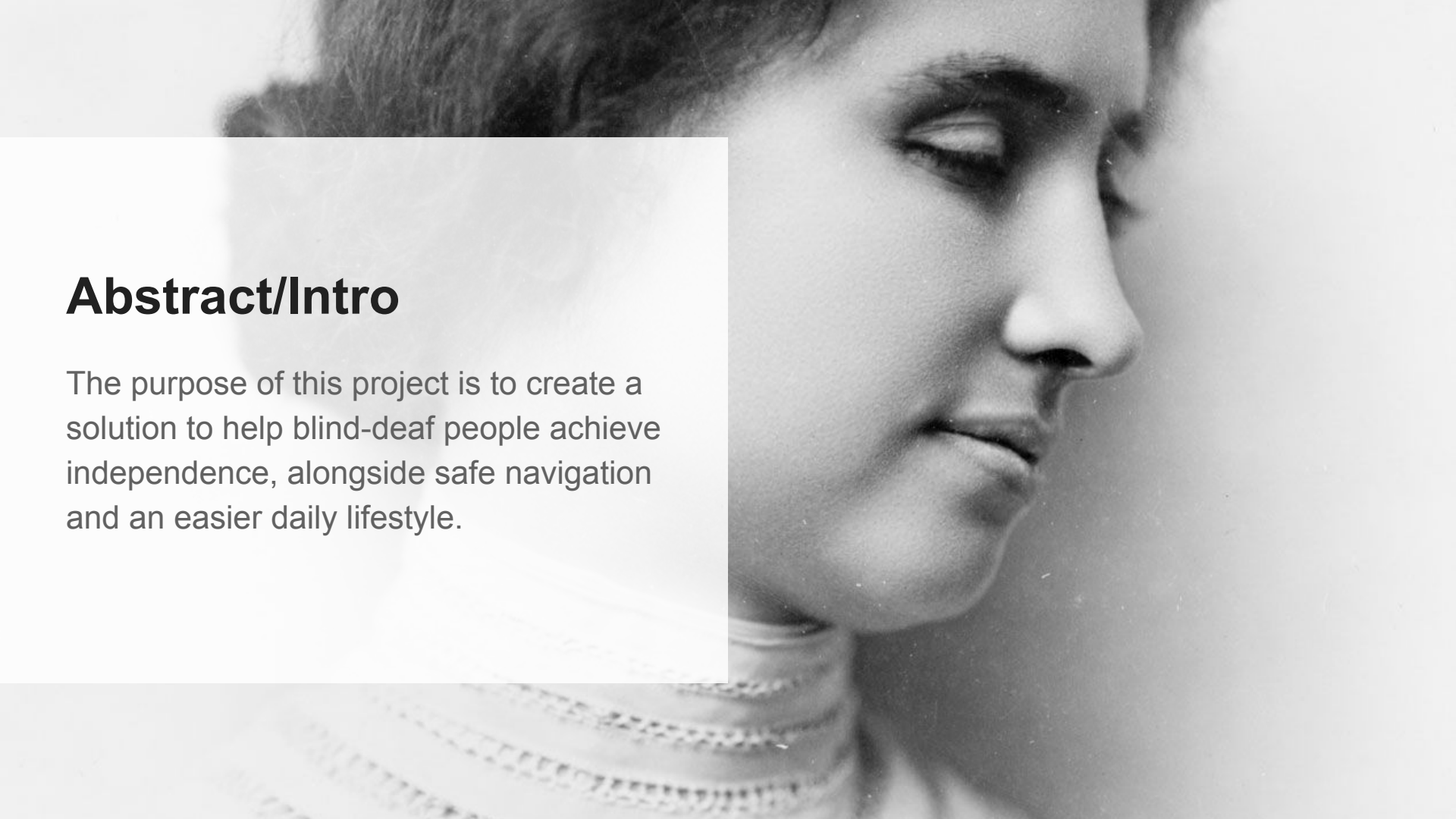
Braille Kit: The Combination Of Braille And Artificial Intelligence To Enhance The Impaired Navigation Experience



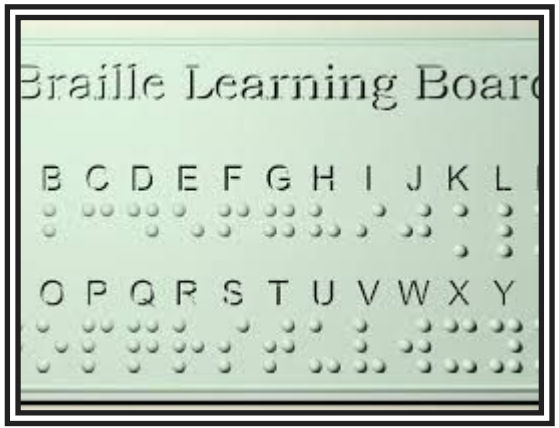
Abstract/Intro

The purpose of this project is to create a solution to help blind-deaf people achieve independence, alongside safe navigation and an easier daily lifestyle.

Little background information

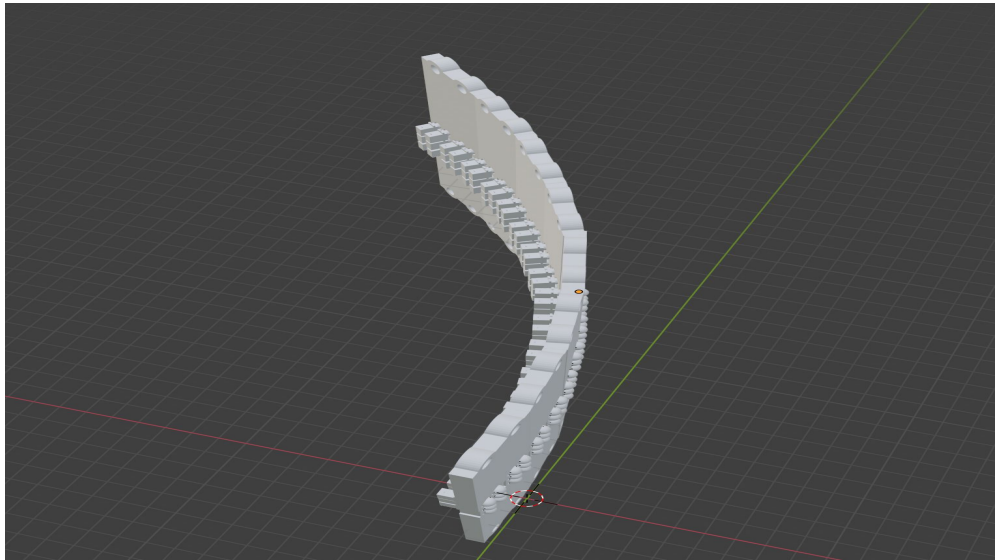
The deaf-blind population in the USA is 50,000 people alone. They do not get the technology and resources they need; ignored in society. Leaving a societal gap between them and the average person.

Braille is a system of touch reading and writing for blind people in which raised dots represent the letters of the alphabet.



Hypothesis

Yes, I can, using computer science, smart systems, a raspberry pi with a camera module, along with an Arduino we can create a system to help the deaf-blind become independent.



The ideation and approach

I’ve realized that there isn’t many people trying to do what I’m doing. There have definitely have been creations of refreshable-braille boards before but this is the first time that anyone is actually attempting to create a system to help them navigate the world.

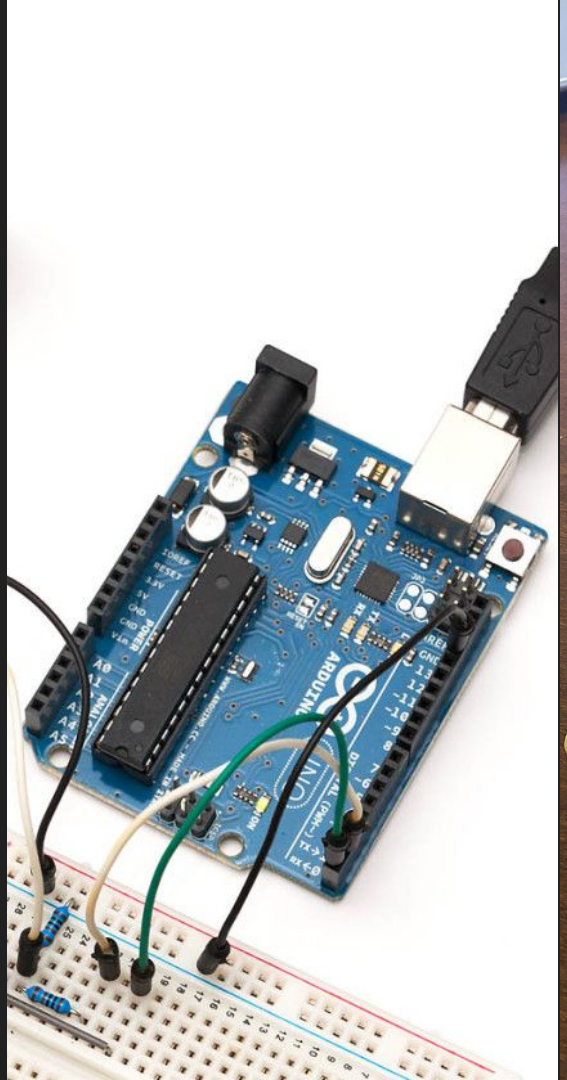
Arduino and Raspberry Pi

Raspberry pi

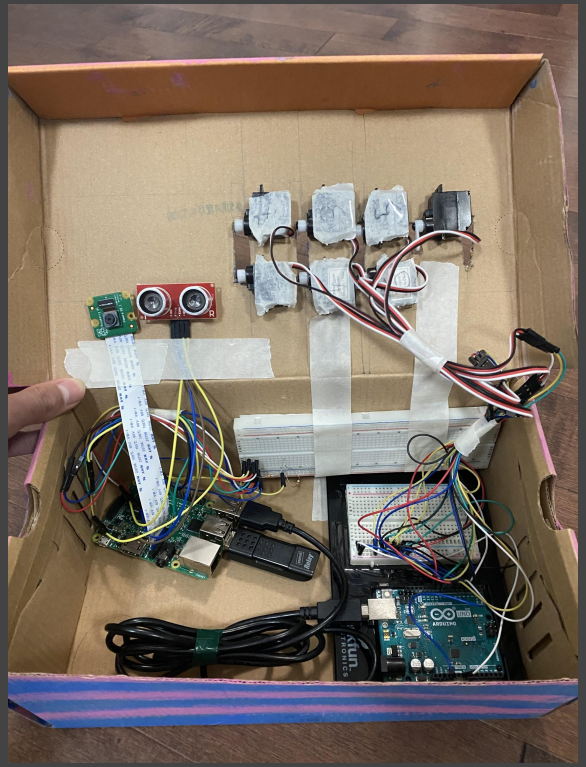
- Hub for system does the computing
- Has Neural Compute Stick
- Connected to Arduino
- Has the ultrasonic sense

Arduino

- Controls the motors



Procedure



1. Choosing our micro-controllers
2. Creation of YOLO model through LabelImg
    a. Set up darkflow on your desktop
    b. Annotate images through LabelImg
    c. Retrain a tiny-yolo model
3. Set up raspberry pi with NCS through Open Vino
4. Set up raspberry pi with the ultrasonic sensor and the pycamera
5. Connect the Raspberry pi with the arduino using pyfirmata
6. Create the code.

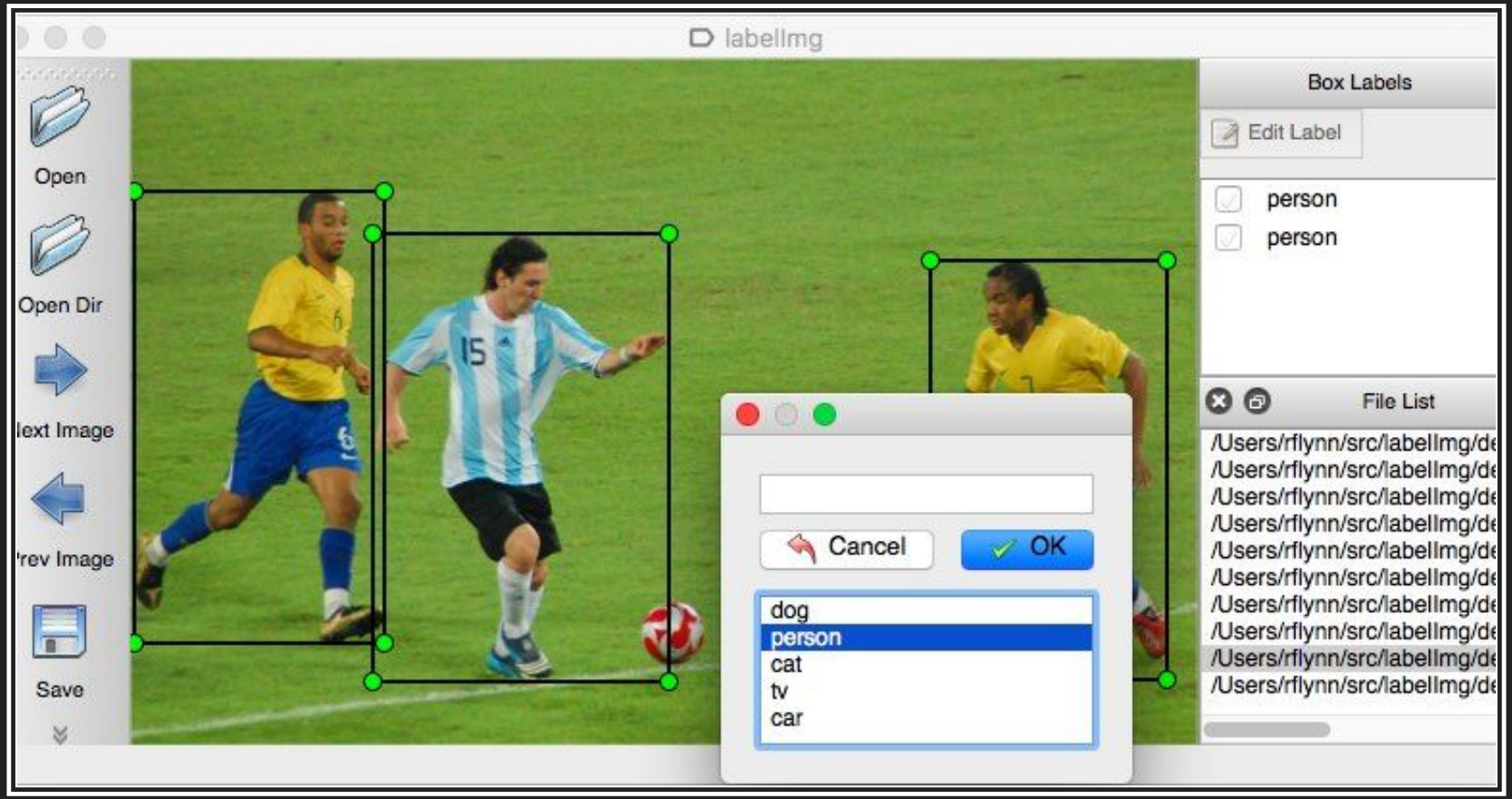
OpenVINO™

TensorFlow

YOLO

As mentioned before we are using an object detection model called YOLO. Using transfer learning I created my own model, hand annotating thousand of images over the course of a week to create an accurate object detection model. It currently detects stop signs, pedestrian hand signals, numbers, and pedestrian signs.

```
dell@hak-ubuntu:~/Bureau/darkflow$ flow --model cfg/yolov2-new.cfg --load bin/yo
lov2.weights --train --annotation train/Annotations --dataset train/Images

Parsing ./cfg/yolov2.cfg
Parsing cfg/yolov2-new.cfg
Loading bin/yolov2.weights ...
Successfully identified 203934260 bytes
Finished in 0.07304739952087402s

Building net ...
Source | Train? | Layer description                | Output size
-------+--------+----------------------------------+---------------
       |        | input                            | (?, 608, 608, 3)
 Load  |  Yep!  | conv 3x3p1_1  +bnorm  leaky      | (?, 608, 608, 32)
 Load  |  Yep!  | maxp 2x2p0_2                     | (?, 304, 304, 32)
 Load  |  Yep!  | conv 3x3p1_1  +bnorm  leaky      | (?, 304, 304, 64)
 Load  |  Yep!  | maxp 2x2p0_2                     | (?, 152, 152, 64)
 Load  |  Yep!  | conv 3x3p1_1  +bnorm  leaky      | (?, 152, 152, 128)
 Load  |  Yep!  | conv 1x1p0_1  +bnorm  leaky      | (?, 152, 152, 64)
 Load  |  Yep!  | conv 3x3p1_1  +bnorm  leaky      | (?, 152, 152, 128)
 Load  |  Yep!  | maxp 2x2p0_2                     | (?, 76, 76, 128)
```


labelImg
Open
Open Dir
lext Image
'rev Image
Save
Box Labels
Edit Label
person
person
File List
/Users/rflynn/src/labelImg/de
/Users/rflynn/src/labelImg/de
/Users/rflynn/src/labelImg/de
/Users/rflynn/src/labelImg/de
/Users/rflynn/src/labelImg/de
/Users/rflynn/src/labelImg/de
/Users/rflynn/src/labelImg/de
/Users/rflynn/src/labelImg/de
/Users/rflynn/src/labelImg/de
/Users/rflynn/src/labelImg/de
Cancel
OK
dog
person
cat
tv
car

```
step 1406 - loss 146.37850952148438 - moving ave loss 154.19132160931767
step 1407 - loss 159.20748901367188 - moving ave loss 154.6929383497531
step 1408 - loss 137.14906311035156 - moving ave loss 152.93855082581294
step 1409 - loss 147.7570037841797 - moving ave loss 152.42039612164962
step 1410 - loss 149.46670532226562 - moving ave loss 152.12502704171123
step 1411 - loss 164.90322875976562 - moving ave loss 153.40284721351668
step 1412 - loss 124.48836517333984 - moving ave loss 150.511399009499
step 1413 - loss 145.4283447265625 - moving ave loss 150.00309358120535
step 1414 - loss 159.3308868408203 - moving ave loss 150.93587290716687
step 1415 - loss 198.9818115234375 - moving ave loss 155.74046676879397
step 1416 - loss 191.8068389892578 - moving ave loss 159.34710399084037
step 1417 - loss 140.72183227539062 - moving ave loss 157.4845768192954
step 1418 - loss 200.97674560546875 - moving ave loss 161.83379369791277
step 1419 - loss 134.82745361328125 - moving ave loss 159.1331596894496
step 1420 - loss 134.74099731445312 - moving ave loss 156.69394345194996
step 1421 - loss 172.9716033935547 - moving ave loss 158.32170944611042
step 1422 - loss 148.45571899414062 - moving ave loss 157.33511040091344
step 1423 - loss 139.74398803710938 - moving ave loss 155.57599816453302
step 1424 - loss 143.20361328125 - moving ave loss 154.33875967620472
step 1425 - loss 150.91415405273438 - moving ave loss 153.99629911385767
step 1426 - loss 164.6321258544922 - moving ave loss 155.0598817879211
step 1427 - loss 155.34124755859375 - moving ave loss 155.0880183649884
step 1428 - loss 156.3590850830078 - moving ave loss 155.21512503679034
step 1429 - loss 123.83740997314453 - moving ave loss 152.07735353042577
step 1430 - loss 105.23585510253906 - moving ave loss 147.39320368763708
step 1431 - loss 138.68060302734375 - moving ave loss 146.52194362160773
```

Working Model

Before

After

Working Model

Before

After

Before

After

Working Model

Before

After

Before

After


HY-SRF05
120203
Y1
HY
SRF05
Vcc
Trig
Echo
OUT
GND
J1
F1
R


Raspberry Pi
J2
v2.1

Speech To Text

People take for granted the ability to verbally communicate. Impaired individuals who are both blind and deaf are faced with one very significant issue: communication. Speech recognition is a viable solution for all of this. Using an open source microphone on the device and speech recognition code the device can understand and convert speech into text which can be converted into braille through motor movements. The Braille Kit uses an audio file recognizer code which converts directions to braille through motor movements.

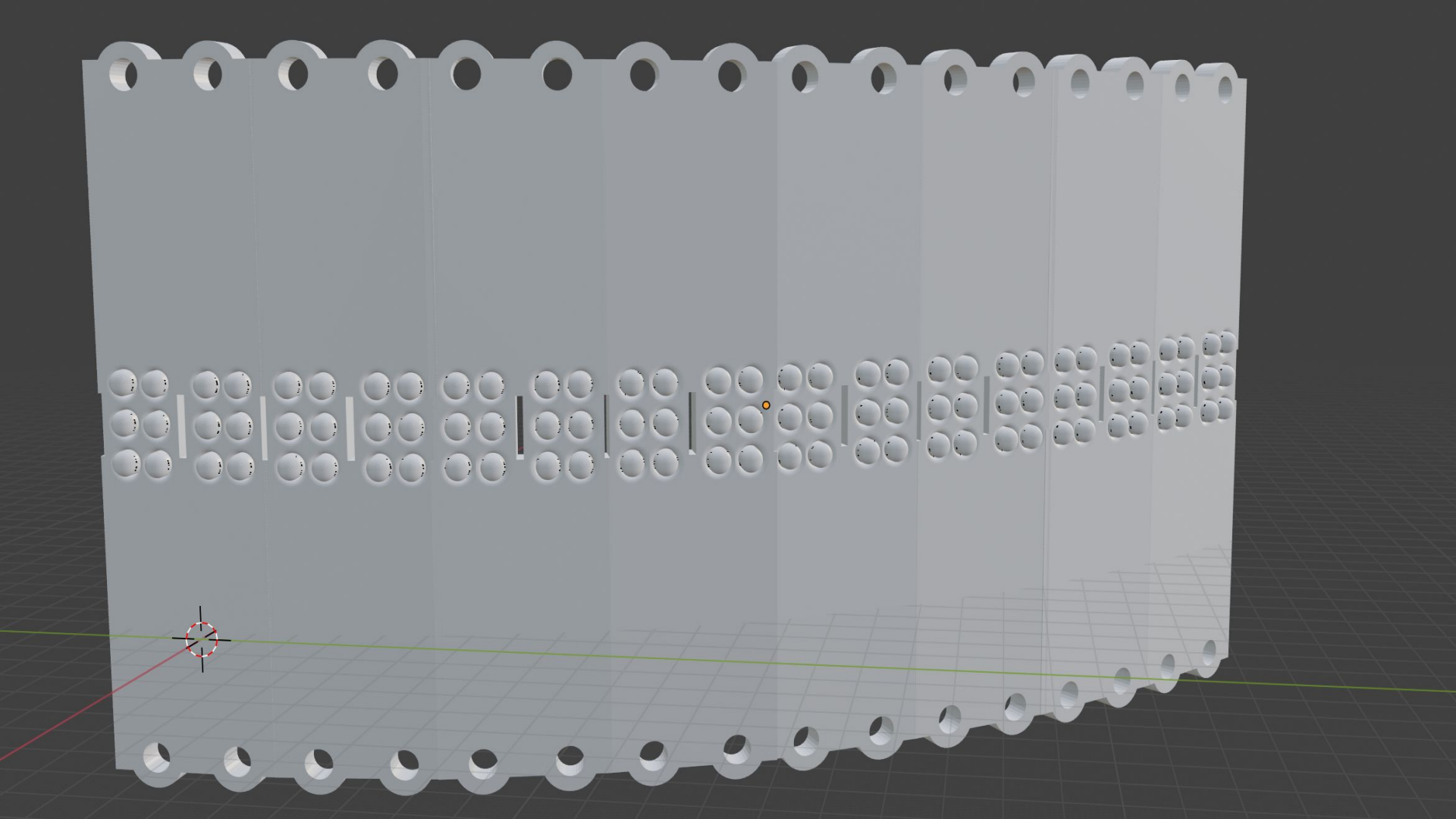
```
import speech_recognition as sr
r = sr.Recognizer()
audio = 'audio.file'

with sr.AudioFile(audio) as source:
        audio = r.record(source)
        print ('Finished')
try:
      text = r.recognize_google(audio)
       print (text)

except Exception as d:
       print (d)
```

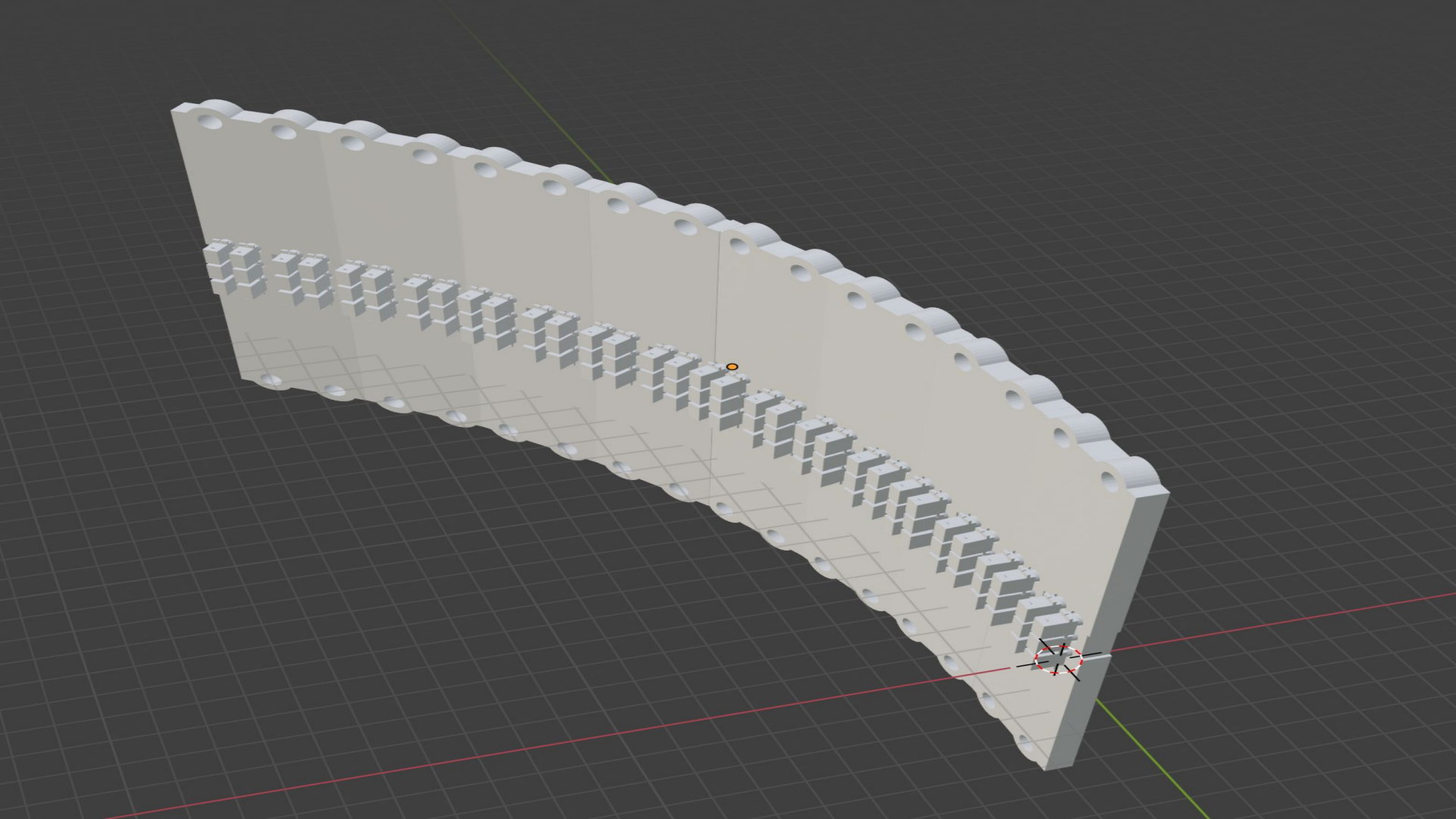
Snippet of Voice Recognition code

Navigation through Google API

Another important functionality of the Braille kit is its vocal use of google maps. I have integrated a google api to allow the user to pull directions to wherever they want to go. The creation and code has been completed however the testing still needs to be done.

3D Model






My Passport

Use and importance

Those in the deaf-blind community can use the final product to help them walk around the world, communicate with others. The goal is for the system to be on a harness around them, with the braille belt around the waist, the camera on their chest, and the arduino and raspberry pi in the fanny pack.

Societal Implications

With this braille board it allows those that are blind-deaf to have more independence for themselves, making it easier for them to interact with others in the world, with less need for translators, reducing the isolation of the deaf-blind community.