Final Presentation

D2: SightMate

by Meera Pandya, Josh Joung, Shakthi Angou

Requirements

Battery Life

minimum of **4 hours** because a guide dog usually takes a break every 4 hours

Accuracy (safety)

at least **70%** because it is the minimum qualification to become a guide dog

Aiming for **80%**

Detection Distance (safety)

minimum of **2 meters** because a user would need enough distance to avoid the obstacle

Weight (health)

no more than **450 grams**, battery pack may be offloaded to waist if needed

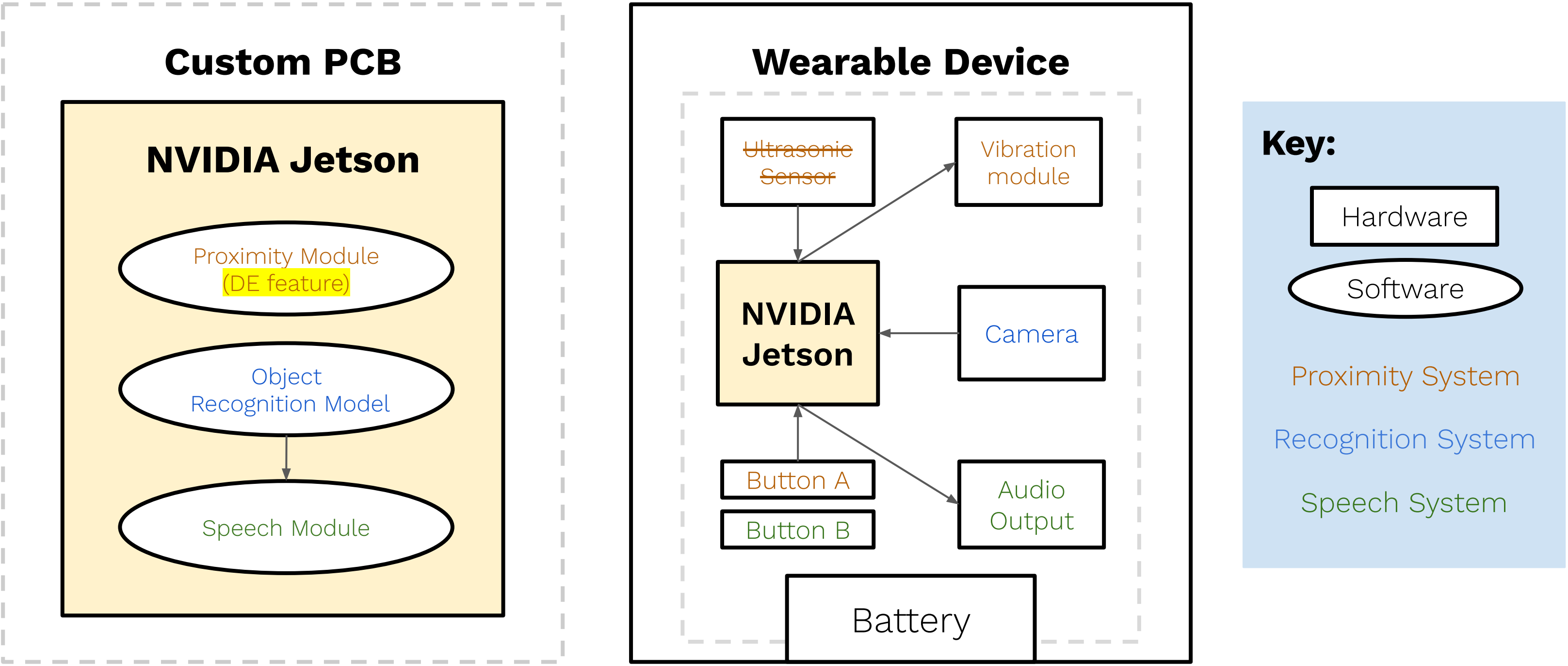
Recognition Delay (safety)

less than **2.5 seconds** to permit 2-meter detection (blind pedestrians walk at .8 m/s)

Noise Detection (safety)

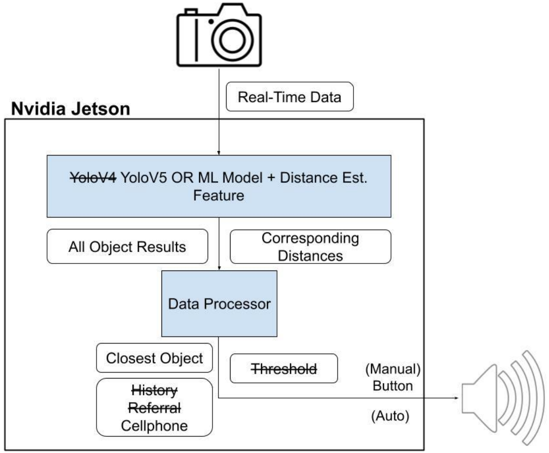
user should be able to **hear surrounding noises** regardless of the audio device

Block Diagram

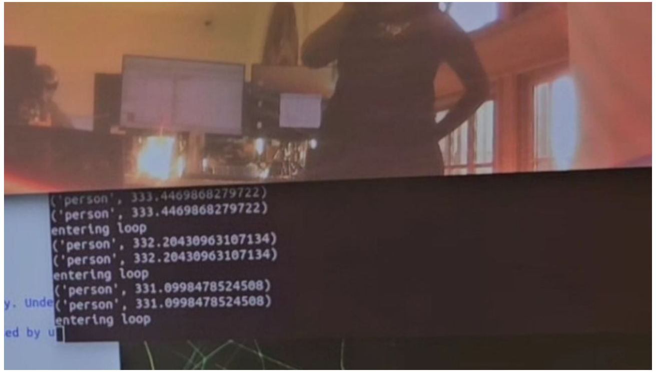


Completed: Object Recognition (OR) Module

- Use ~~YoloV4~~ YoloV5 OR ML Model
- Created **Distance Estimation** module (using reference images)
- Use pandas to interpret reference data (.csv) for 5 objects
- Detect & est. distance every frame




```
chair_test7.jpg
[{'name': 'chair', 'boxwidth': 95.2259521484375, 'confidence': 0.867239773273468}, {'name': 'chair', 'boxwidth': 104.465576171875, 'confide
nce': 0.839117109775432}, {'name': 'chair', 'boxwidth': 121.91455078125, 'confidence': 0.8363508582115173}, {'name': 'chair', 'boxwidth':
147.88623046875, 'confidence': 0.8148535490036011}, {'name': 'chair', 'boxwidth': 39.742431640625, 'confidence': 0.7098137736320496}, {'nam
e': 'chair', 'boxwidth': 114.846923828125, 'confidence': 0.674306929114807}, {'name': 'chair', 'boxwidth': 53.232177734375, 'confidence':
0.5866798758506775}, {'name': 'chair', 'boxwidth': 56.770751953125, 'confidence': 0.4514462053775787 4}, {'name': 'chair', 'boxwidth': 69.53
369140625, 'confidence': 0.4455329477787018}, {'name': 'chair', 'boxwidth': 89.156982421875, 'confidence': 0.4395681917667389}, {'name': 'd
ining table', 'boxwidth': 291.574951171875, 'confidence': 0.2717465162277221 7}]
Object: chair Boxwidth: 95.2259521484375 Distance: 480.0302785187882 cm
Object: chair Boxwidth: 104.465576171875 Distance: 437.5732371094507 6 cm
Object: chair Boxwidth: 121.91455078125 Distance: 374.9457307524401 cm
Object: chair Boxwidth: 147.88623046875 Distance: 309.0980153266572 cm
Object: chair Boxwidth: 39.742431640625 Distance: 1150.1898209294468 cm
Object: chair Boxwidth: 114.846923828125 Distance: 398.0197188428041 cm
Object: chair Boxwidth: 53.232177734375 Distance: 858.7163305647155 cm
Object: chair Boxwidth: 56.770751953125 Distance: 805.1917362266861 cm
Object: chair Boxwidth: 69.53369140625 Distance: 657.3984410659738 cm
Object: chair Boxwidth: 89.156982421875 Distance: 512.7062299588978 cm
ref: chair_test7.jpg object: chair
```

Completed: Proximity & Speech Modules

- Device control buttons:

  Control A: vibration setting (alerting user non-disruptively)

  Control B: (Single Press): speech identification of immediate obstacle (MVP)

  (Double Press): continuous speech identification setting

- Proximity Module:

  ~~**Ultrasonic sensor** picks up on objects within 2m of the user.~~ DE feature in the OR model is connected to the **vibration motor** placed at the back of the user's neck to alert them of obstacles approaching.

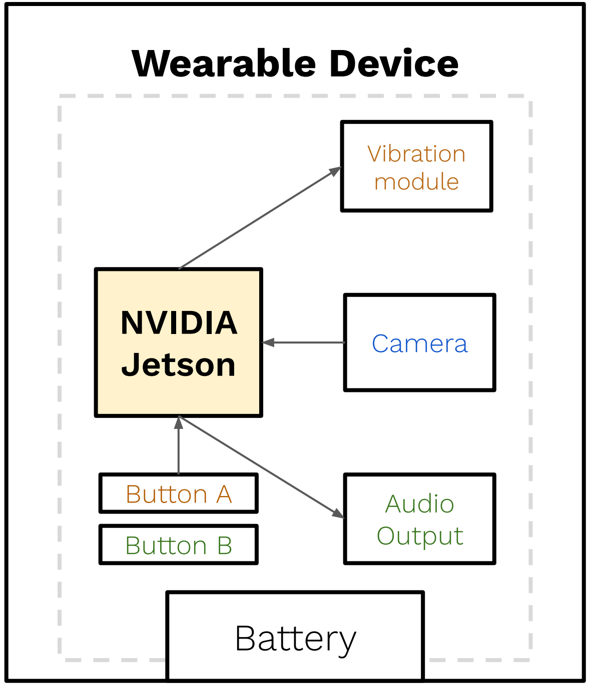
- Speech Module:

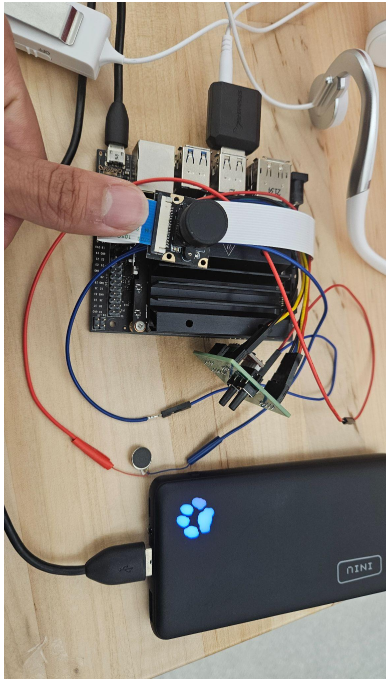
  Output from the OR model is processed and converted into speech using a **TTS engine** called **espeak** that we use with the **pyttsx3** python library.

Completed: Hardware Implementation

System	Components	Integration Plan
Object Recognition System	e-CAM50_CUNX/NANO Camera	MIPI interface w/ onboard connector
Proximity Detection System	~~HC-SR04 Ultrasonic Sensor~~ DE feature in OR model	Connect to Jetson GPIO pins via custom PCB for voltage conversion and/or current limiting
	Vibration Motor	
Dictation System	Control Buttons	
	CM108 Audio Converter	USB plug-in
Overall	10,000 mAh Battery	Rechargeable power bank

Complete Solution





Test, Verification and Validation

Testing	Verification	Risk Mitigation
Object recognition model	**Identify the closest** object using a built-in camera	Implement the model with Yolov7 for greater accuracy
Distance estimation module	**Compare the distance** of the closest material measured by the model with the actual distance	
Text-to-speech module	**Pass noise testing** that tests whether users can hear both speech and background noise	Bone-conduction headphones
Vibration module	**Vibrate** if there is an object within 2m in front of the user	Recalibrate the distance estimation feature
Device controls (buttons)	Turns on and off (vibration module, auto/manual settings) when the **user presses** the button	Unit-testing for the different modes
Module integration	-**Compare the time** for the product to provide the result to the minimal recognition delay. -The overall product **weighs less than 450g** -Battery lasts more than **4 hours**	Find bottlenecks in our system to improve latency
Functionality	The device **detects and alerts the closest object** within 2m of range from the user	All of the above

Test Success

Testing	Metrics	Result
Object Recognition Model	> 70% on identifying an object	**95%** (38/40 images, 5 objects)
Distance Estimation Feature	± 30cm of actual object distance	Tests done on 4 different distances. Average of 21.5% uncertainty **within ± 30cm**
Text-to-speech Module	user-testing for surrounding sounds 20 trials each object	**100%** (20/20 person, 20/20 couch, 20/20 chair, 20/20 cat, 20/20 cellphone)
Vibration Module	> 95% accuracy on vibration	**100%** (20/20 on person, 20/20 on nothing)
Device Controls (buttons)	100% accuracy on controls	**100%** (20/20 on button A, 20/20 on button B)
Module Integration (weight)	< 450g on the overall product weight	**192g** (device) **+ 209g** (battery) **= 401g** < 450g

Test Failure & Improvement Plan

Testing	Metrics	Failure	Plan
Recognition Delay	< 2.5s to recognize an object	~8 seconds delay for 20 seconds testing. Frame delay due to the latency of the OR model	Reduce the bottleneck of the model -restrict the number of classes to identify) -skip a detection every 3 frames
Battery Life	> 4 hours of running	Headless device needs to be deployed	Deploy the device headless and run it on the battery
Functionality	> 70% on the accuracy	Headless device needs to be deployed	Measure accuracy with the headless device with LAMP volunteers

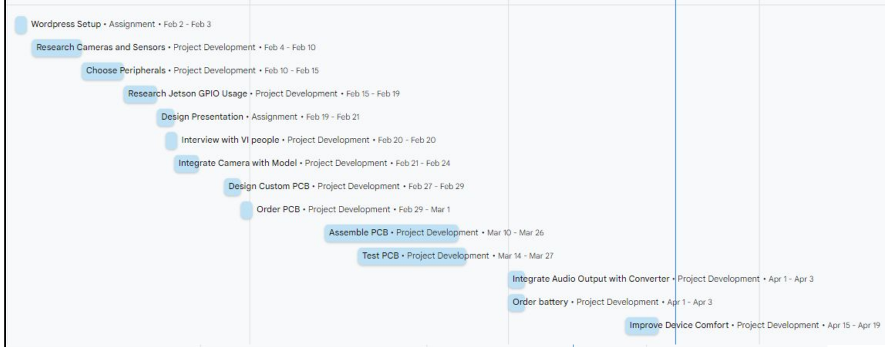
Trade-offs

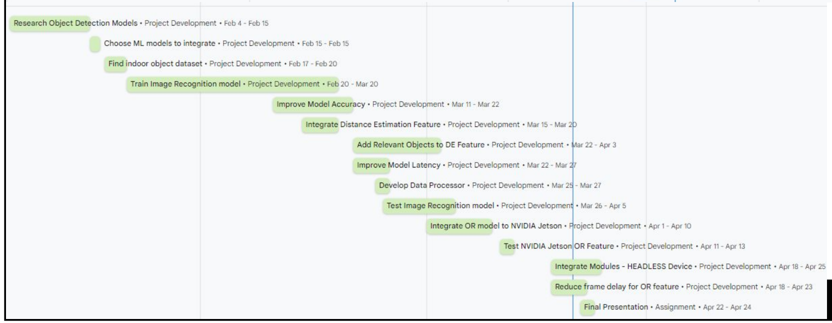
- Chose **pre-trained model** instead of trained model with indoor object dataset

Model	Real Objects	Detected	Falsely detected	Percentage (%)
Pre-trained	58	49	5	84.4
Trained	58	21	4	36.2

- Chose **Distance Estimation feature** in the OR model instead of ultrasonic sensor
  - Ultrasonic sensor does not work well with Jetson Nano
  - DE feature rarely goes over ± 30cm, although **some calibration is necessary**

Actual (m)	Detected (m)	Off (m)
1.80	1.82	+ 0.02
1.20	0.89	- 0.31
0.20	0.38	+ 0.18
2.2	1.94	- 0.26





Timeline

Workload Split:

Hardware by **Meera**

Object Recognition Model by **Josh**

Speech and Vibration Modules by **Shakthi**

Overall Integration and Device Design by **All**

To do:

1. Headless Device
2. User Testing
3. Reduce frame delay
4. Improve product appearance

Project Takeaways

Technical knowledge/technology necessary for the project

1. Machine Learning techniques, frameworks (Josh)
2. Circuit & PCB design, GPIO configuration (Meera)
3. Linux Sound Architecture, Sensor & GPIO Configuration (Shakthi)
4. NVIDIA Jetson Nano configuration/programming (All)

Importance of slack time

1. Integration of multiple modules took longer than expected
2. Progress not met as expected (conflict with other courses or events)

Reading documentation / Research

1. Github README & Issues
2. NVIDIA tutorials
3. Stack Overflow
4. YouTube videos