The Self-Driving Human

Team D3: Proposal Presentation
William Shaw, Max Tang, Andrew Wang

Use Case

CHALLENGE: Visually impaired pedestrians rely on external aids to cross the road.

- Not all crossroads have reliable or maintained aids
- Obstacles (like debris) on the road can further reduce safety

SOLUTION: A head-worn device that aids the user in crossing the road by providing real time visual-to-audio guidance.

- “WALK” vs. “DON’T WALK” signals
- Helps user stay within the crosswalk
- Alerts to unexpected objects in the walking path

Alternative existing solutions:

- Accessible pedestrian signals - not widely implemented
- Guide robots/dogs - too expensive

ECE Areas:

- Software: Two ML models used to detect walk signs, avoid obstacles, and stay on the crosswalk
- Hardware: High-resolution camera to capture data, Jetson Nano for inference, GPS for positioning, and headphones for interfacing

Use-Case Requirements (1)

System 1: Walk Sign Image Classification Model

Metric	Requirement	Justification
AUROC	0.9 (the model ***correctly*** ranks random “GO” higher than random “STOP” 90% of the time)	A realistically achievable metric for distinguishing between “STOP” and “GO” that is sufficiently high
Majority Vote Accuracy	99% accuracy when taking majority vote over 10 frames	Each frame 90% correct ➔ probability that 6/10 frames are wrong is <1% Must be as high as possible to ensure safety
Inference Latency	3 seconds after walk sign changes from “STOP” to “GO”	0.3s to process each frame Users can still safely cross after waiting 3s
Audio Latency	0.1 seconds	Users should hear audio cue to cross as soon as model output is finalized

Use-Case Requirements (2)

System 2: Crosswalk Obstacle Detection Model

Metric	Requirement	Justification
Multi-Class AUROC	0.75 across all objects detected	A realistically achievable metric for YOLO models that is sufficiently high
Majority Vote Accuracy	99% accuracy when taking majority vote over 5 frames	Each frame 90% correct → probability that 3/5 frames are wrong is <1% Must be as high as possible to ensure safety
Inference Latency	1 seconds after an obstacle enters camera's range for model to work	0.2s to process each frame Gives users enough time to reach
Audio Latency	0.1 seconds	Users should hear cue to cross as soon as model finishes

System 3: Crosswalk Navigation

Allowed Deviation	Detect when user veers more than 20 degrees off-course from other side of road	20 degrees is a wide enough angle to allow the user some freedom when walking
Speed and Clarity of Audio Feedback	Audio feedback clearly guides user to correct their direction within 1 second	1 second is fast enough for a user to correct their direction without having strayed too far off

Technical Challenges

Systems 1 & 2: Walk Sign Image Classification Model + Crosswalk Obstacle Detection Model

- Test and tune models on real world data to meet classification accuracy metrics
- Make models lightweight enough to fit on Jetson Nano microcontroller by quantizing parameters
- Make models fast enough to meet inference speed requirements

System 3: Crosswalk Navigation

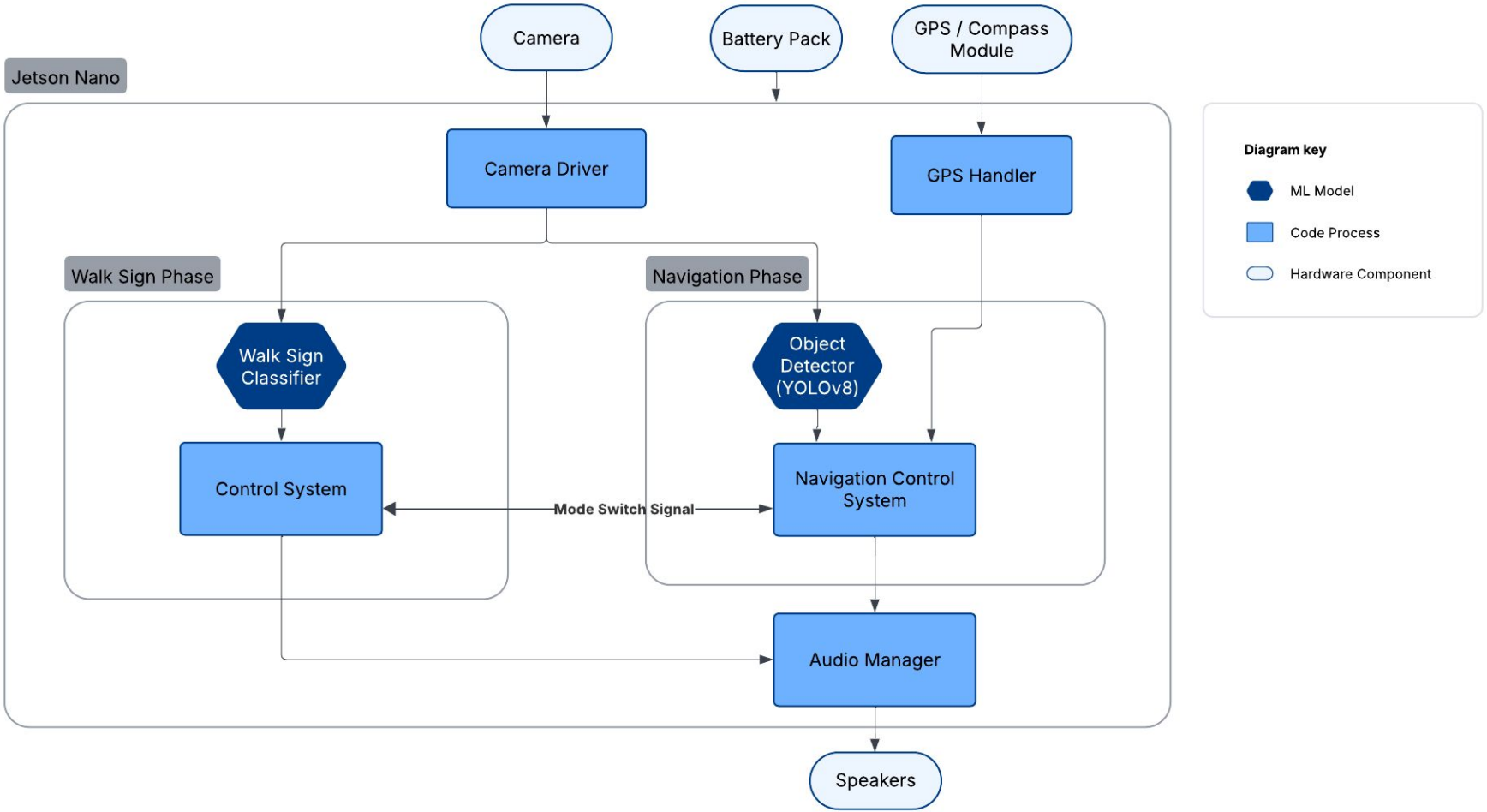
- Make position tracking accurate and processing fast enough to alert user of any deviations from crosswalk
- Distinguish if an object on the sidewalk is an obstacle or not

Solution Approach - Hardware

- NVIDIA Jetson Nano
  - Run on-board ML for reduced latency and independence from network conditions
  - Portable enough to be mounted to a person
  - Higher power consumption vs other microcontrollers, but needed for running ML models on real-time video data.
- Camera Module - Arducam IMX219
  - Wide FOV (175°) lets us see as much info as possible (walk sign, cars, crosswalk)
- GPS Receiver / Compass
  - Lets us track user position and orientation while crossing road
- Audio output - Koss Porta Pro
  - On-ear speakers connected via the 3.5mm jack on the board
  - Needs to not block ambient noise from the road (safety)
- Battery - 10,000mAh battery
  - Expect around 5-10W of power draw from all components
  - Gives us around 4 hrs of battery

Solution Approach - Software

- “Walk Sign Classifier” ML Model
  - Smaller CNN to detect “WALK”, “DON’T WALK”, and ‘Not a crosswalk’
  - Will use existing datasets as well as collected images to train
- “Crosswalk Obstacle Detection” ML Model
  - Fine-tunable “off-the-shelf “ YOLOv8 model to detect cars and obstacles
- Run our code in two phases, one for each model
  - Having two separate phases reduces model and code complexity
  - Easier to debug and manage
- Python backend to read from our modules (GPS, camera, etc)
- Audio feedback done through Python using Text-To-Speech library (pyttsx3)


Camera
Battery Pack
GPS / Compass Module
Jetson Nano
Camera Driver
GPS Handler
Walk Sign Phase
Navigation Phase
Walk Sign Classifier
Object Detector (YOLOv8)
Control System
Mode Switch Signal
Navigation Control System
Audio Manager
Speakers
Diagram key
ML Model
Code Process
Hardware Component

Testing, Verification and Metrics (1)

System 1 and 2: Walk Sign Image Classification Model + Crosswalk Obstacle Detection Model

- Per-frame classification accuracy test
  1. Collect real-world dataset from different crosswalks in different visibility conditions
  2. Measure performance metrics such as AUROC
- Majority vote accuracy test
  1. Run model inference on 10 consecutive frames in test set and compare majority vote output with actual label
  2. Measure accuracy as (no. correct predictions / no. total predictions)
- Response time test for inference and audio feedback
  1. Measure total inference time for 10 frames and audio playback delay
- User verification test
  1. Test if a user wearing the helmet can orient the camera to detect the walk signal and obstacles in crosswalk

Testing, Verification and Metrics (2)

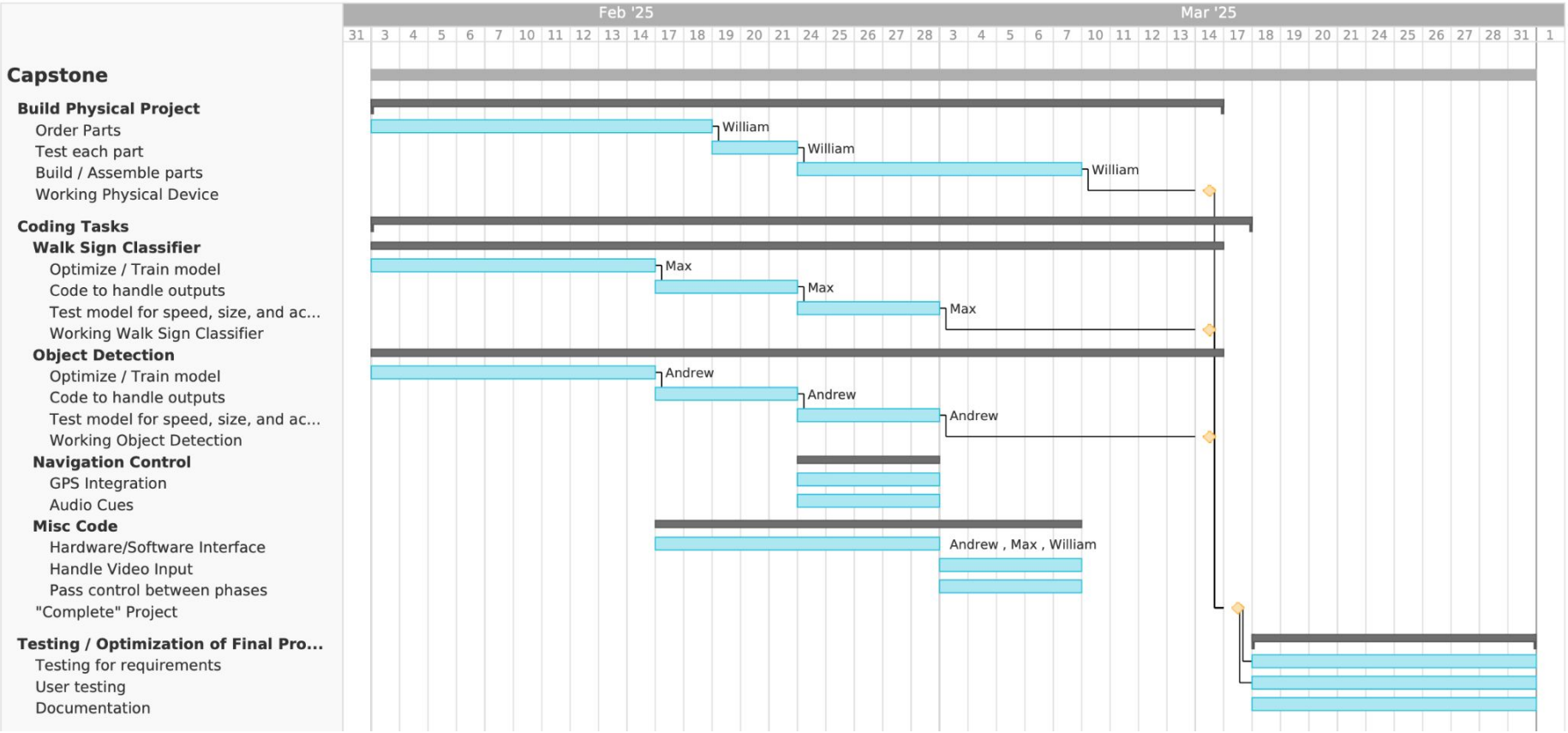
System 3: Crosswalk Navigation

- Veering angle detection test - accuracy of detecting a 20 degree deviation
  1. Have testers walk along a straight line and at controlled angles
  2. Measure accuracy as how often the system is able to detect 20+ degree angle deviations (no. of detections / no. of runs)
- Audio response time test - time from deviation detection to audio feedback
  1. Simulate veering >20 degrees and measure the time until audio feedback is played
- Audio feedback clarity
  1. Have testers walk off course and test audio feedback
  2. Survey testers if feedback was able to help them walk in straight line

Tasks and Division of Labor

William	Max	Andrew
Hardware / Physical Build ● Ordering parts ● Unit testing each part ● Assembling final build Handle user orientation ● Organizing live testing Code: ● Hardware/Software Integration	Train “Walk Sign Classifier” ML model ● Finding/collecting datasets ● Optimizing and training ● Testing model requirements Write code for handling walk sign detection ● Quantizing the model ● Integrating the model into the final build	Train “Object Detector” ML model ● Finding/collecting datasets ● Optimizing and training ● Testing model requirements Write code for handling walk sign detection ● Quantizing the model ● Integrating the model into the final build

Schedule


Feb '25
Mar '25
31 3 4 5 6 7 10 11 12 13 14 17 18 19 20 21 24 25 26 27 28 3 4 5 6 7 10 11 12 13 14 17 18 19 20 21 24 25 26 27 28 31 1
Capstone
Build Physical Project
Order Parts
Test each part
Build / Assemble parts
Working Physical Device
Coding Tasks
Walk Sign Classifier
Optimize / Train model
Code to handle outputs
Test model for speed, size, and ac...
Working Walk Sign Classifier
Object Detection
Optimize / Train model
Code to handle outputs
Test model for speed, size, and ac...
Working Object Detection
Navigation Control
GPS Integration
Audio Cues
Misc Code
Hardware/Software Interface
Handle Video Input
Pass control between phases
"Complete" Project
Testing / Optimization of Final Pro...
Testing for requirements
User testing
Documentation
William
William
William
Max
Max
Max
Andrew
Andrew
Andrew
Andrew , Max , William