GateGuide

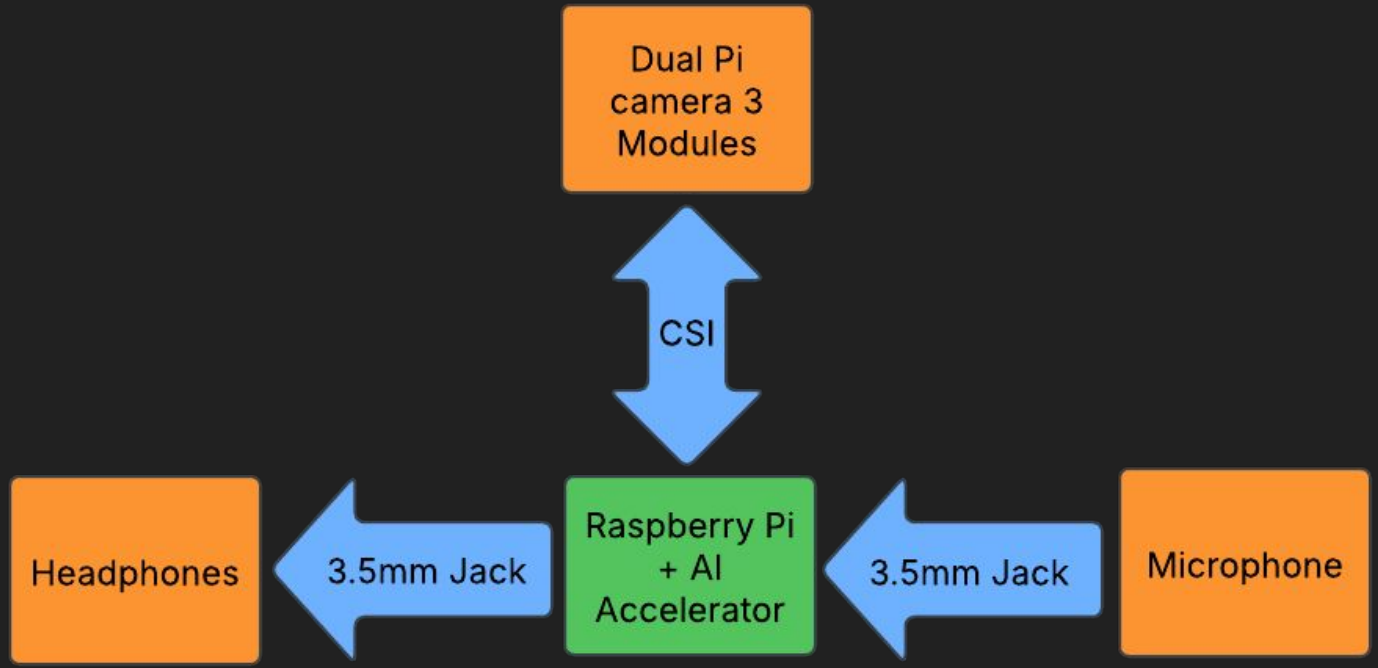
Providing real-time navigation assistance with auditory feedback

Team E4: Krrish Jain, Opalina Khanna, Daniel Kim

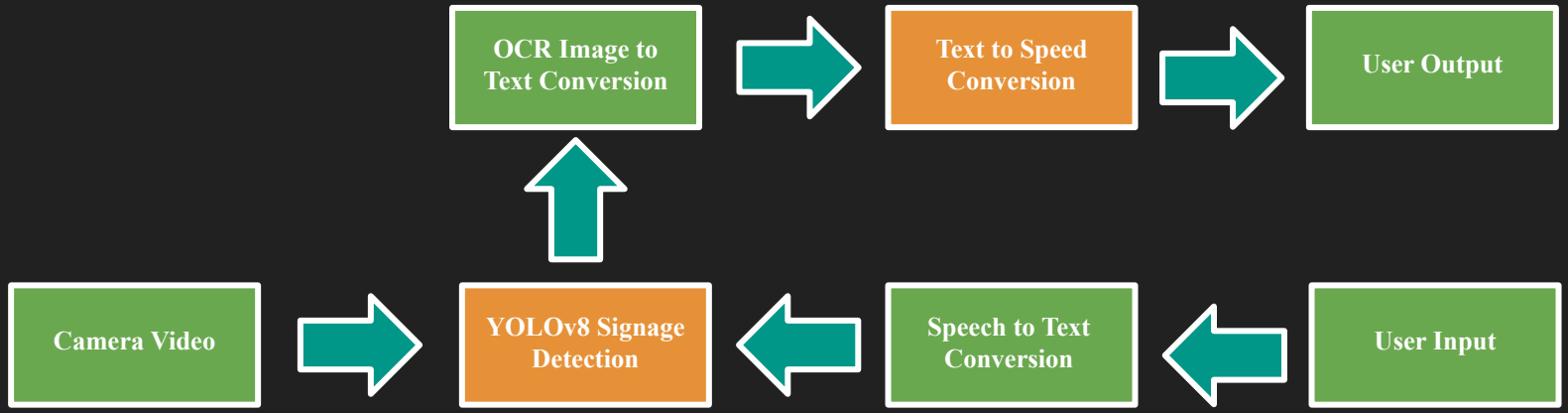
Use Case Requirements

Latency <2s	Time between video feed, processing, and client output should be fast in order to ensure efficient direction
Signs (Accuracy >90%)	Accuracy of the YOLO model that identifies the direction of the user's destination
Gate Number (Accuracy >95%)	Accuracy of the OCR model to identify when the user has reached their final gate
Battery Life (5 hours)	Should be enough time to get from security to the final gate
Weight (<2kg)	Device should be lightweight for long uses

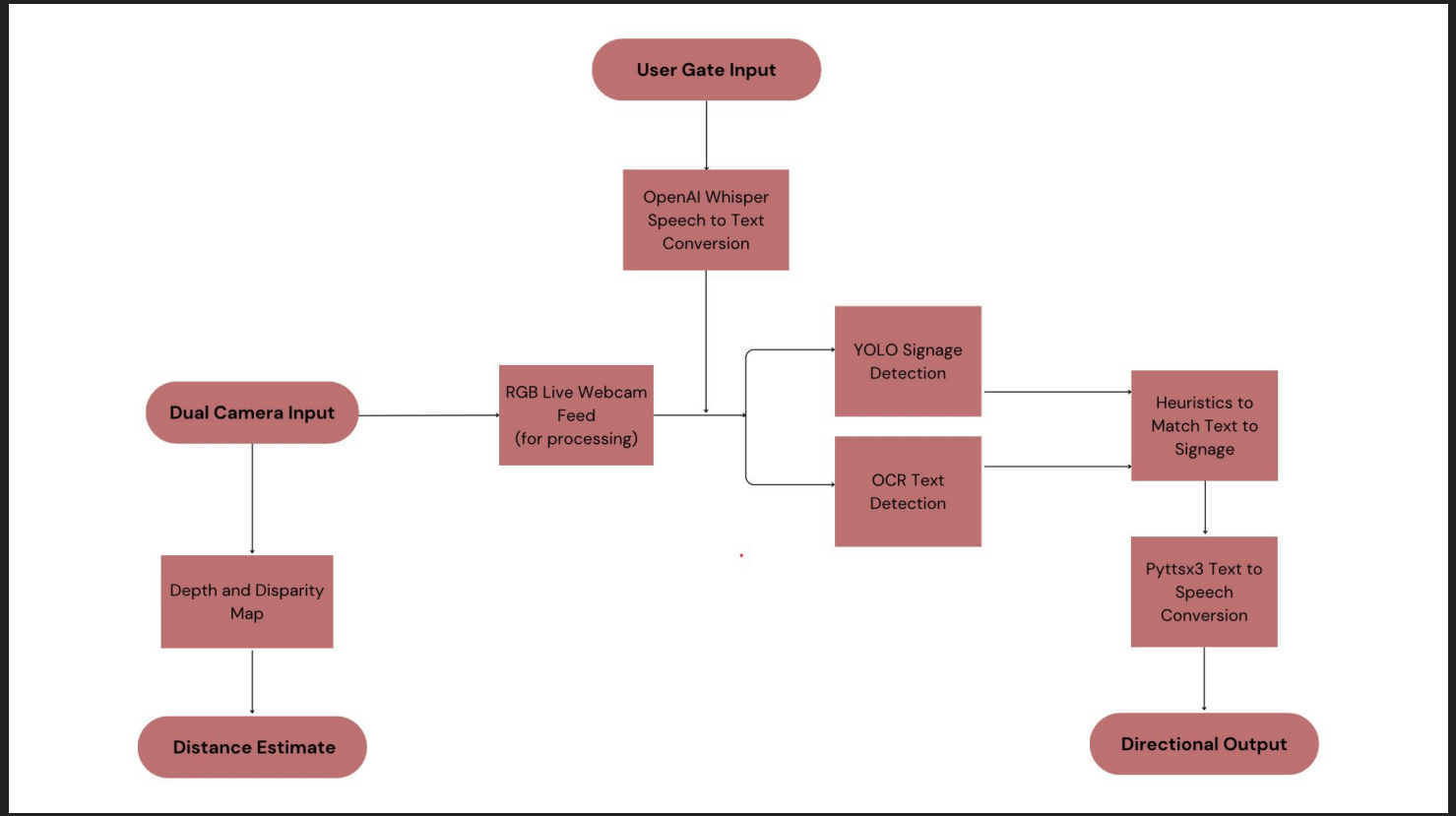
Hardware Block Diagram



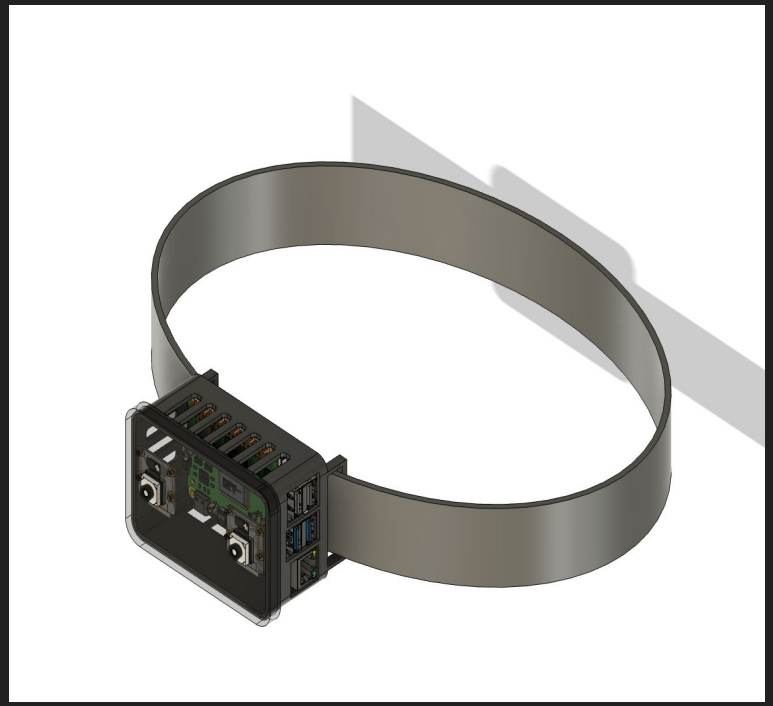
Software Block Diagram



Complete Solution

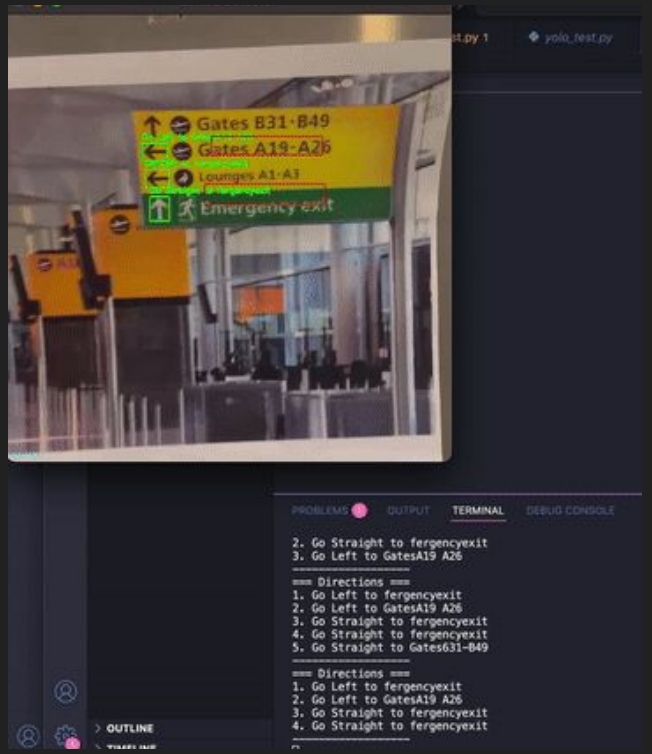


Complete Solution



Final Demonstration

- Video demonstration of the working, belt-mounted device, transmitting live webcam feed
- Printed signage in an indoor environment to simulate an airport terminal
- Laptop speaker to register user input and output feedback
- Monitor to display depth map and RGB video footage



0.5x speed

Testing, Verification, and Metrics

Latency (Time between video input and client feedback - 20 trials)	• Already tested on laptops, current latency is less than 100 ms per frame (running at 15 fps) • Working on compiling the custom models onto the Raspberry Pi to test end-to-end latency • As of now we have a latency of 300-400 ms on device
Signs (Accuracy of YOLO model - 117 images)	• Tested on laptop - achieved 80% for directional arrows
Text Detection (Accuracy of OCR model - 117 images)	• Tested on laptop - 90% accuracy for gate number detection
Battery Life (3 trials)	• Using the 10,000mAh Anker Power Bank gives us a battery life of about 4 hrs on testing
Speech-to-Text and Text-to-Speech (Accuracy of Whisper and pyttsx3 - 40 trials)	• Tested with USB Mic on Laptop - 60% accuracy achieved, need to fine tune our scripts to maximize this
Weight (<2kg)	• 3D-printed case for the device, and electronics weighs <150g + 250g for the battery

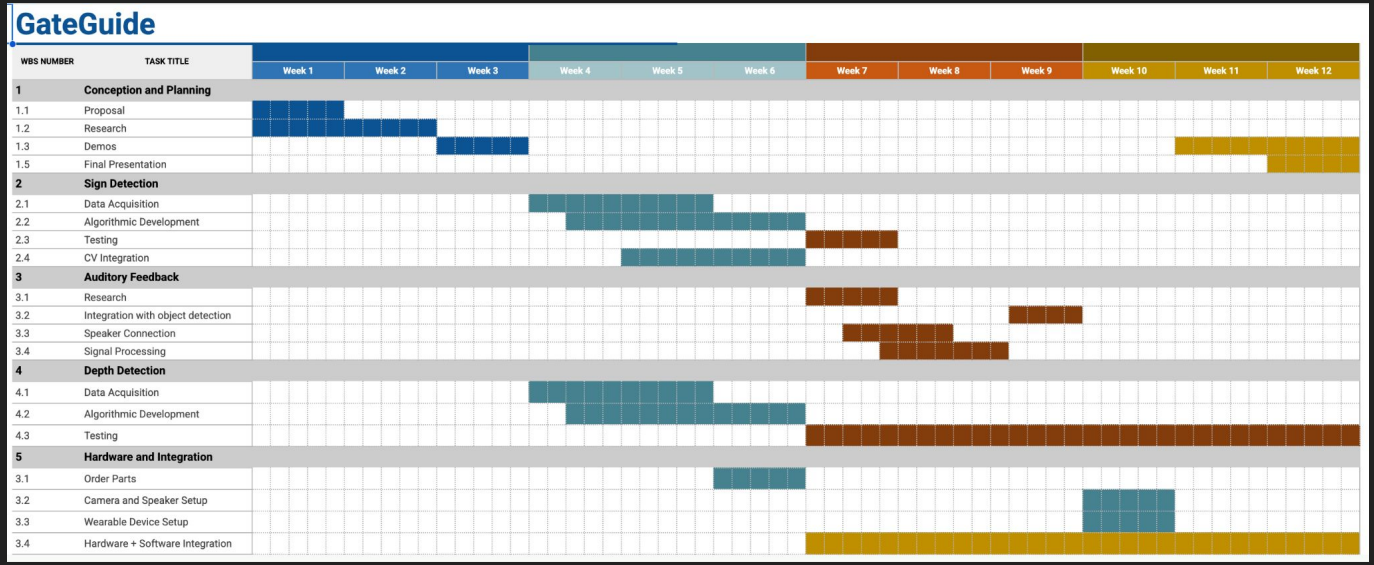
Design Trade-Offs

Design Component	Chosen Solution & Rationale	Alternatives & Why Not
Compute Unit	● Raspberry Pi with Hailo8 AI accelerator ● We underestimated how much compute power we'd require and therefore have a higher latency ● Ultimately choose the pi because it's more portable and power efficient only requiring 10-12W ● We are currently getting about 6 FPS, however, we have some integration left and expect that to drop	● Jetson Nano is more powerful with a dedicated GPU. ● It has more compute power and would give us a lower latency ● It would consume 25W which is almost double what the Pi requires.
Camera	● 2 pi cameras in a stereo camera setup ● Faster frame processing since supports native encoding: 60 FPS ● Worse depth Accuracy	● eYs3D depth camera ● Slow frame processing from USB: 10-12 FPS ● Better Depth accuracy since it has IR tof sensors

Design Trade-Offs

Design Component	Chosen Solution & Rationale	Alternatives & Why Not
YOLO Model	• Used a nano, lightweight version of YOLO (with fewer classes) in order to ensure efficiency on the Pi • Achieved 80% accuracy with <1s latency with YOLOv8n (nano)	• YOLO full size - 90% accuracy and >2s latency with the full size model • This model did not meet our initial use case requirements, leading us to prioritize latency over accuracy
OCR	• Rapid OCR - Increased FPS by 3x • Running 1 time every 5 frames - Reduced processing time from >200ms to 50ms	• Easy OCR - Accuracy of 90%, but dropped processing to 5fps • Running OCR on every frame - waste of resources, since subsequent frames are usually similar / identical
STT and TTS	• OpenAI's Whisper with over 95% accuracy as an offline model	• Vosk wouldn't even recognize our voice so we stopped using it

Project Management



Changes in Project Schedule:

- Added integration time
- Added unit and end-to-end testing time
- Delay in wearable device and speaker setup
- Speech to Text model needed to be redone, delaying the UI integration

Lessons Learned

- Integrate early!! We should have started integrating much earlier. Things go wrong as we port scripts from our laptops to the Pi, as we change cameras, etc. Some of these could have been avoided if we integrated earlier.

- Communication is key. We need to keep our teammates updated on what we're doing, it ensures that everyone is on the same page and we can help each other is someone is falling behind.