D3: The Self-Driving Human

Authors: William Shaw, Max Tang, Andrew Wang
Affiliation: Electrical and Computer Engineering, Carnegie Mellon University

***Abstract*—Visually impaired pedestrians have difficulty crossing streets, especially when crosswalks lack accessible pedestrian signals such as audible walk indications. The Self-Driving Human system is a chest harness equipped with cameras and other modules to assist visually impaired pedestrians by alerting them to changes in the walk signal and avoid obstacles in the crosswalk, allowing them to cross streets safely and confidently.**

***Index Terms*—Audio Feedback, Camera, Crosswalk, Ergonomic Design, GPS, Image Classification, Machine Learning, Navigation, Object Detection, Pedestrian, Power Supply, Real-Time Feedback, ResNet Model, Safety, Visually Impaired, Walk Sign Detection, YOLOv12 Model**

1 INTRODUCTION

Navigating urban environments presents a significant challenge for visually impaired pedestrians. Traditional aids such as canes and guide dogs offer some assistance, but they do not always provide sufficient situational awareness, particularly when crossing streets. Many crosswalks lack accessible signals, and obstacles like debris or cars can further increase risks. As a result, there is a critical need for a solution that enhances safety and autonomy for visually impaired individuals when crossing roads.

Our proposed solution is a chest-worn device that leverages real-time visual-to-audio guidance to assist users in safely navigating crosswalks. By detecting "WALK" and "DON'T WALK" signals, monitoring the user's alignment within the crosswalk, and alerting them to unexpected obstacles, this device enables visually impaired pedestrians to navigate crosswalks with higher independence and confidence.

Current assistive technologies, such as tactile paving and auditory signals at crosswalks, are often inconsistent and unavailable in many locations. Mobile applications with GPS-based navigation offer limited real-time environmental awareness. In contrast, our approach integrates a high-resolution camera, an inertial measurement unit (IMU), and real-time machine learning (ML) processing on a Jetson Orin Nano to deliver precise, immediate guidance. This method eliminates reliance on external infrastructure, ensuring usability across various environments. The primary goal of this project is to create a reliable, responsive, and user-friendly system that significantly improves pedestrian safety for the visually impaired.

2 USE-CASE REQUIREMENTS

To ensure the system effectively aids visually impaired pedestrians in crossing streets safely, the following use case requirements have been established with public health, safety, and welfare, as well as global, cultural, social, environmental, and economic factors considerations in mind:

1. **Scope and camera speed of walk sign and obstacle detection:** The chest harness should be able to detect walk signs and crosswalk obstacles in a 105° field of view at 30 frames per second. This ensures that the system can process visual data in as close to real-time as possible while still being integrated into a mobile device, providing accurate and timely guidance to the user even in dynamic urban environments.

2. **Crosswalk deviation angle:** The system should be able to keep the user from deviating more than 45° from the line of the crosswalk. This ensures that the user can safely cross the street without wandering into the path of stopped cars.

3. **Device battery life:** The chest harness should be able to last at least 2 hours on battery. This ensures that users can wear it without frequent recharging, making it suitable for daily commutes and extended outdoor use. The system's power consumption must be optimized to balance performance and efficiency, and users do not have to worry about the chest harness dying while walking.

4. **Accuracy of walk sign classification per frame:** The system must correctly classify the "WALK" signal with an Area Under the Receiver Operating Characteristic Curve (AUROC) of at least 0.9 to ensure reliable decision-making. This accuracy is calculated per frame input passed into the object detection model, and was chosen to be a high accuracy to maximize pedestrian safety.

5. **Accuracy of walk sign classification over 5 frames:** The probability of misclassifying the walk signal over five consecutive frames should be less than 1%. This accuracy is measured by taking a majority vote over the 5 predictions, and was chosen to be as high as possible to both ensure user safety and also mask any false positives from a single frame's output. Each frame is sampled after the previous frame is processed by the image classification model.

6. **Time to output prediction:** The system must produce an auditory walk signal indication within 3 sec-

onds of the signal changing in the real world. This time includes all model inference time and producing the audio. This requirement was chosen to give the image classification model ample time to process 5 frames, but is also fast enough to give the user time to cross the street at a reasonable speed. Real walk signals often display the "WALK" signal for 3-5 seconds.

7. **Accuracy of crosswalk object detection model:** The model should detect obstacles in the user's path with at least 90% accuracy, reducing risks from tripping over obstacles or bumping into cars. The detection system must function effectively in various environmental conditions, including low-light scenarios and crowded urban areas. It should also be able to distinguish between static obstacles that the user could collide with and dynamic objects such as other walking pedestrians that might not actually impede the user's path.

8. **Clarity and volume of user feedback:** The feedback provided as a result of the crosswalk obstacle detection model should be clear and instructive enough to help the user avoid obstacles. This metric is measured by conducting user surveys with visually impaired people and collecting their ratings on various instructions, allowing us to create a system that caters to their specific needs. The volume should also not mask the environment so that users can still be aware of their surroundings.

9. **Ergonomic Design and Comfort:** Both the chest harness and external battery pack should be lightweight and balanced to make sure that prolonged usage is not uncomfortable or straining for the user. This includes chest harness padding and adjustable straps for different body proportions, and earbuds that are comfortable.

3 ARCHITECTURE AND/OR PRINCIPLE OF OPERATION

The Self-Driving Human provides navigation assistance by integrating computer vision, object detection, IMU-based heading tracking, and real-time audio feedback. As such, the system architecture requires a mix of hardware and software components. These are combined to detect walk signs, track user alignment to a crosswalk, and provide audio cues for navigation.

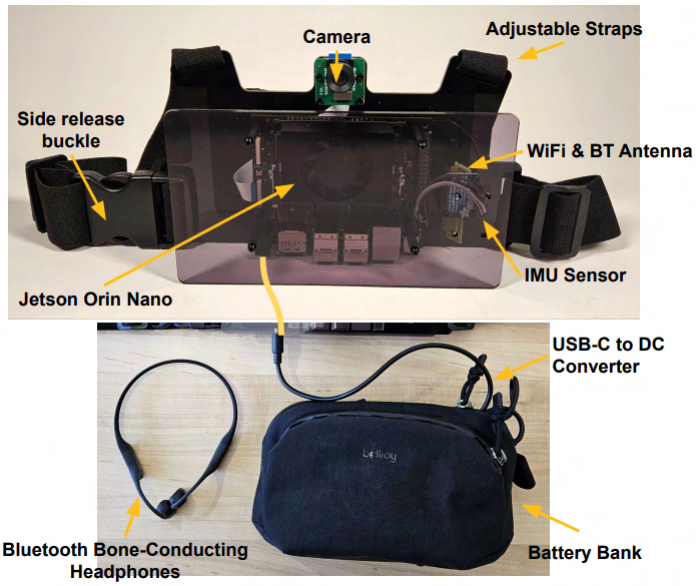

Figure 1: Diagram of system.

Hardware Architecture:

1. **Jetson Orin Nano:** The processing unit that runs our ML models and code for walk sign classification, object detection, and crosswalk alignment.

2. **Camera Module (IMX219):** The camera used to capture real-time video data. This data is fed into our ML models to detect walk signs and obstacles. Our model has a wide FOV of 105°(D) and supports multiple resolution/framerate combinations, though we are currently using 1280x720p@30fps.

3. **IMU Sensor (BNO055)**: 9-DOF sensor with a MEMS accelerometer, magnetometer, and gyroscope. Allows us to get the absolute orientation of the user to determine their heading to $\pm$ 2 degrees even when stationary. This is used for our crosswalk alignment.

4. **Headphones:** We had two options for types of headphones: on-ear and bone conducting. From an implementation standpoint, both function essentially the same. However, since on-ear headphones may limit how much a user can hear from their surrounding and reduce their spatial awareness, bone-conducting headphones would help increase safety. As such, we chose to use the wireless AfterShokz OpenMove (bone-conducting) headphones.

5. **Power Supply:** We used a 24,000mAh powerbank. To supply the required power to the Jetson Orin Nano, we use a USB-C PD to 15V 5A DC Converter. This allows us to use a standard power bank, rather than using a more niche DC power bank.

Software Architecture:

1. **Walk Sign Classification Subsystem:** This subsystem outputs a boolean "Safe to Cross" signal based on live video feed from camera. The video data is processed with a ResNet-based classifier [2], predicting whether or not each frame has a "WALK" or "DON'T WALK" signal. If the classifier outputs 5 consecutive "WALK" detections, the control system passes control to the next Navigation Subsystem.

2. **Navigation Subsystem:** This system outputs real-time audio feedback to guide the user across the crosswalk. It raises alerts if the user deviates $> 45\,\text{deg}$ from the intended path, or if an unexpected obstacle on the sidewalk is detected.

3. **Audio Manager:** Cues from both of the subsystems are passed into an Audio Manager, which can output Text-To-Speech (TTS via pyttsx3) based cues or simple audio alerts from an audio library. The Audio Manager ensures that the most critical notification is being played, ensuring the user's safety. Audio cues were user tested for clarity and ease of use.

Principle Of Operation:

The Self-Driving Human operates in two major software phases. The Walk Sign Classification Phase and the Navigation Phase. As implied, the Walk Sign Classification Phase runs a ResNet model to classify the pedestrian crossing light. There are three possible states: "WALK", "DON'T WALK", and "NONE". For the system to detect a valid crossing signal, five consecutive "WALK" signals need to be detected. Once this occurs, the system prompts the user to begin walking. At this point, control is passed to the Navigation Phase. At the start, a measurement is taken for the estimated position of the crosswalk endpoint, based on visual data data and IMU measurements. Then, as the user is crossing the crosswalk, the IMU measurements are used to ensure that the user stays aligned to the desired path. Simultaneously, a YOLOv12 [5] model continuously monitors for unexpected obstacles in the path of the user. Based on these, the Navigation Phase outputs commands to the user, helping them to avoid obstacles and stay on the sidewalk. Cues from both the Walk Sign Classification Phase and the Navigation Phase are sent to an Audio Manager Process, which determines what audio cues to play based on priority. Fig. 2 includes a block diagram of this system.

3.1 Changes from Design Review Report

The primary design change since the design review was the shift from a head-worn device to a chest-worn device. The motivation for this decision was that users are more likely to turn their head to face directions that are not exactly the same direction as the one in which they are walking in. Since the calculation of crosswalk deviation relies on the user pointing the device at their destination, then mounting the IMU sensor to their chest would result in better deviation detection. Another change was that we report deviations of more than 45°, rather than 20°. We did so as we found that 20° was too narrow a value to constitute a meaningful deviation from the crosswalk path.

4 DESIGN REQUIREMENTS

Each use-case requirement spawns a corresponding technical design requirement. By meeting the following design requirements, the system ensures that it can likewise satisfy the use-case requirements.

1. **Scope and camera speed of walk sign and obstacle detection:** The camera must be able to capture video frames at a rate of 30 frames per second and a resolution of 720p with a 105° field-of-view. This frame rate is fast enough to keep up with the model inference speed, while the resolution and field-of-view is large enough to match the images in the datasets used to train both the image classification and object detection models.

2. **Crosswalk deviation angle:** The IMU sensor must be accurate enough to detect variations in headings with a granularity up to 1°. This accuracy is high enough to allow the system to calculate the angle between the direction the user is facing and the endpoint.

3. **Device battery life:** The battery used to power the system must provide at least $6,000$ mAh. This is sufficient to provide 2 hours of battery at the required power to run the Jetson Orin Nano at the highest performance setting, which is necessary during the model inference phase. This calculation can be found in Section 5.3.

4. **Accuracy of walk sign classification per frame:** The walk sign classification system utilizes the ResNet image classification model to classify the "WALK" vs "DON'T WALK" vs "NONE" signal. The model must achieve an AUROC of at least 0.9 on the test dataset to ensure reliable decision-making. The model is trained on a diverse dataset of crosswalk signals under various lighting and weather conditions. This ensures the model is robust enough for real-world scenarios, balancing accuracy with computational feasibility while maximizing user safety.

5. **Accuracy of walk sign classification over 5 frames:** The model must correctly classify at least 90% of individual frames to ensure that the probability of 3/5 consecutive frames being misclassified is $< 1\%$. This condition is automatically met if the previous design requirement regarding the accuracy of a single frame is satisfied. By sampling non-consecutive frames, we can assume that each frame is independent and that the number of incorrectly predicted frames

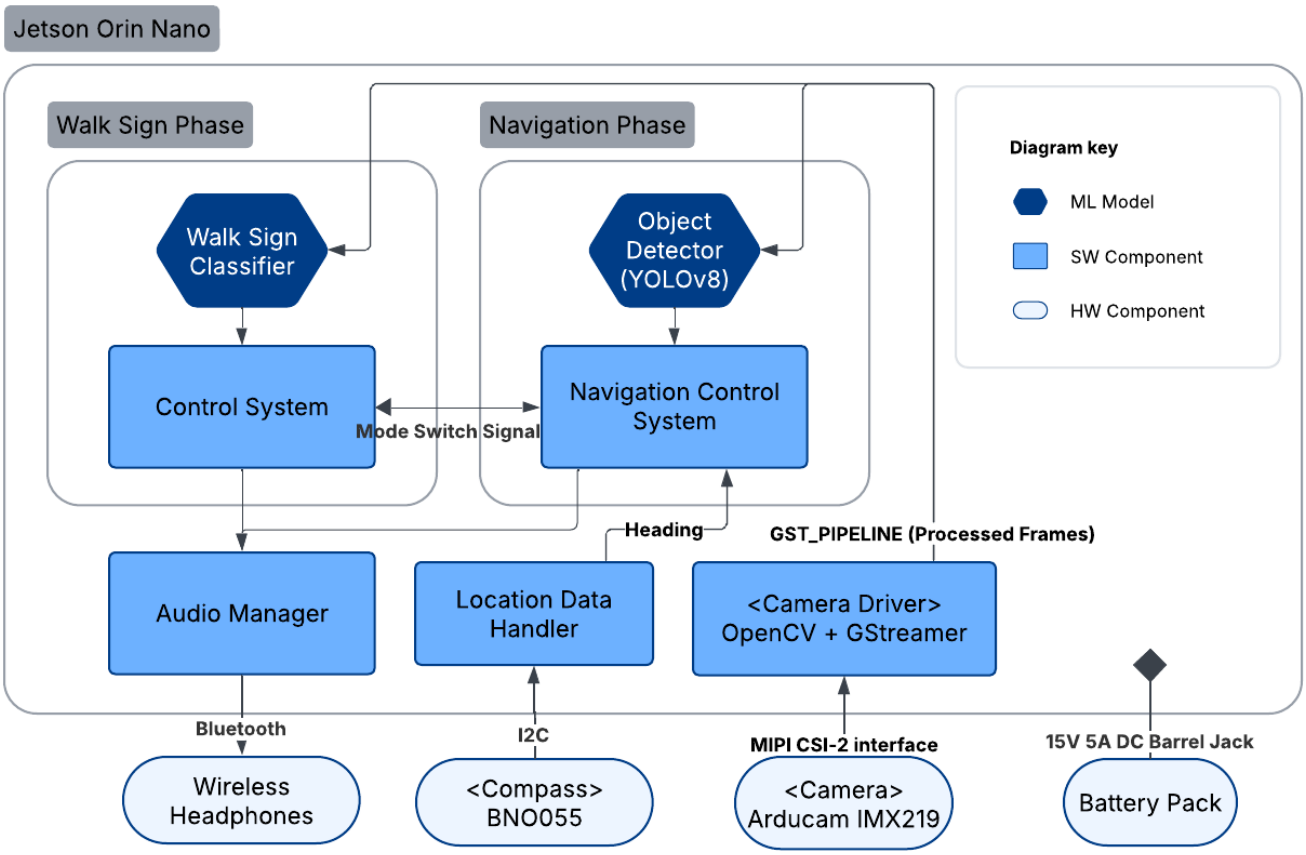

Figure 2: The overall block diagram for The Self-Driving Human. Larger version can be found in Fig. 6.

follows a binomial distribution:

$$P(X = k) = \binom{5}{k}(0.1)^k(0.9)^{5-k} \quad (1)$$

Thus the probability of outputting an incorrect majority vote is $P(X \geq 3) = P(X = 3) + P(X = 4) + P(X = 5) = 0.00856$, which satisfies the requirement. This reduces the impact of individual frame misclassifications, improving system robustness and user trust in the guidance system.

6. **Time to output prediction:** The ResNet image classification model must classify the walk signals with an inference latency of ≤ 3 seconds for all 5 frames. This can be satisfied by quantizing the off-the-shelf model and ensures that users receive timely guidance, preventing delays that could impact safe crossing decisions.

7. **Accuracy of crosswalk object detection model:** The object detection model should also achieve a minimum accuracy of 90% in detecting obstacles in the user's path. The model is based on an off-the-shelf YOLOv12 architecture, fine-tuned using an online dataset of various intersections and crosswalk environments. The model should be able to differentiate between static obstacles (ex. curbs, cars) and dynamic obstacles (ex. other pedestrians). The detection system should function effectively in crowded urban environments where the number of dynamic obstacles is high and in weather scenarios with poor visibility.

8. **Clarity and Volume of User Feedback:** The feedback produced by the crosswalk obstacle detection system should be clear, concise, and actionable. The feedback should not obscure the user's ability to hear environmental sounds, such as approaching vehicles or nearby pedestrians. Instructions were chosen and recorded based on user study results.

9. **Ergonomic Design and Comfort:** The chest harness, which includes the camera and microcontroller, should weigh no more than 700 grams and have an even weight distribution to prevent discomfort and strain. The battery pack should not weigh more than 2 kg and should fit inside a fanny pack, which can be worn comfortably around the user's waist or slung across their shoulder.

5 DESIGN TRADE STUDIES

5.1 ML Model Complexity and Inference Latency

One of the primary design choices is the complexity of the CNN-based architecture used for object detection and walk sign classification. More complex models like YOLO or ResNet models offer higher accuracy but come with increased computation time, which can be problematic for real-time pedestrian guidance. On the other hand, lightweight models are faster and more power-efficient but may sacrifice detection accuracy.

We consider that high-complexity models (ResNet, YOLO) exhibit high accuracy in detecting walk signs under diverse conditions, but notably have increased inference time, potentially causing delays in user feedback. Additionally, we note that they have higher power consumption, potentially reducing battery life. Lightweight models (MobileNet SSD, Tiny-YOLO) exhibit comparatively low latency and have lower computational requirements, making them suitable for edge devices. However, their major drawback is reduced accuracy, leading to potential misclassification of walk signs, or incorrect detection of objects in the road.

We determine that accuracy is of higher importance, and so we decided to use a higher complexity model. However, we looked into methods to reduce the computational burden, such as quantization or neural network pruning to reduce the effective size of the model and preserve battery life while retaining good detection/classification performance. In this way, we directly address the drawbacks of using more complex, heavyweight models while benefiting from the comparatively good performance.

5.2 On-Device vs Remote Inference

We considered the trade-off between processing the object detection using on-device inference versus offloading computations to a smartphone or cloud service.

On-device processing gives lower latency (enabling quicker real-time predictions) and has no reliance on internet connectivity. The main drawback of this method is that it is limited by the device's computational power. Remote processing (e.g., sending frames to a smartphone app or cloud) can use more advanced models for improved accuracy, and offload processing from the wearable, reducing power consumption. However, this method introduces network latency, which can delay user feedback and require an active internet or Bluetooth connection. Ultimately, we decided to use on-device processing due to real-time latency constraints and the need for independence from external connections. Additionally, this removes the extra variable of implementing a potentially slow Bluetooth module, removing unnecessary complexity from our project.

5.3 Processing Unit Selection

The Jetson Orin Nano must provide sufficient processing power to run deep learning models in real time while keeping power consumption manageable for wearable use. The Jetson Orin Nano was selected over alternatives like the Raspberry Pi 4 due to its superior computational performance. While it consumes more power (15W vs. 5W for Raspberry Pi 4), its GPU acceleration enables real-time processing of ML models. We acknowledge that this choice

comes with a higher energy consumption/power requirement, and plan to adapt our other software components to accommodate this. Given that the power consumption is now higher, we selected a 24,000mAh power bank to balance runtime (2 hours) and weight. The Jetson Orin Nano draws 7-15W depending on power mode, and so our worst case analysis is as follows:

$$(15W * 2hrs * 1000)/5V = 6,000mAh \quad (2)$$

5.4 Input and Sensor Selection

The camera must provide a high enough resolution for object detection and walk sign classification while maintaining a sufficient frame rate for real-time inference. The relationship between bandwidth (B), resolution (R), and frame rate (F) can be modeled as

$$B = R * F \quad (3)$$

The IMX219 (1080p @ 30fps) was chosen as a balance between image clarity and real-time processing. Alternatives like the IMX477 (12MP) offered higher resolution but reduced frame rate. We believe that this camera offers the best balance between frame rate, resolution, and user comfort to enable our ML models to give accurate feedback on clear images to the user.

We determined that using the Google Maps API was not necessary for navigation within the scope of our proposed project, and so we decided to use an IMU sensor (BNO055) in order to provide orientation information to our navigation module. In this way, we may ensure that the user remains on course during the crossing period.

5.5 Audio Output

Audio latency is critical for real-time feedback and is affected by the transmission method. We may represent the relationship between latency (L), packet size (P), and bandwidth (B) as

$$L = P * B \quad (4)$$

We determined that a Bluetooth connection would introduce too many variables in audio feedback, such as connectivity issues, potentially high bandwidth causing latency saturation, and so the USB sound card was chosen over Bluetooth due to lower latency and fewer connectivity issues, despite requiring a wired connection. We do plan on polling users on the comfort of needing a wired connection to determine user satisfaction with this design choice.

Audio instructions provide clear, detailed guidance but can be difficult to hear in noisy environments. Haptic feedback (e.g., vibrations) is more discreet and reliable in all environments but may not convey complex information as effectively as voice instructions. We decided to use audio instructions out of interest for clarity and ease of understanding for the user, which are conveyed through on-ear headphones as opposed to through a speaker. We debated transmitting audio directly through a speaker as well, but out of interest for audio clarity and ease of understanding for the user, we decided to first test our headphone setup in order to remove a possible source of difficulty for the user. Once again, we poll the users on the benefits and drawbacks of using audio feedback via on-ear headphones as opposed to the other methods discussed.

6 SYSTEM IMPLEMENTATION

6.1 Hardware Implementation

The Jetson Orin Nano serves as the central processing unit responsible for running our ML models. It handles the computational load of image processing, walk sign classification, and navigation tasks in real time. We reason that it is capable of facilitating quick, real-time inference with our trained ML models while being portable enough for the user to comfortably carry. The IMX219 camera module is responsible for capturing real-time video data. This camera has a wide field of view (FOV) of 105° and supports multiple resolution/frame rate combinations. Our implementation operates at a resolution of 1280x720 at 30fps, which is a sufficiently high resolution for accurate obstacle detection and classification of walk signs. Additionally, the frame rate should be high enough to ensure that we are surveying the surroundings at an appropriately frequent rate. The BNO055 IMU sensor we selected is a 9-degree-of-freedom (DOF) sensor that includes an accelerometer, magnetometer, and gyroscope. It provides absolute orientation data with an accuracy of ± 2 degrees, even when stationary. This sensor is essential for ensuring that the user maintains the correct alignment while crossing the street. With regards to audio feedback modality, we used an on-ear bone conducting headphone that can connect directly to our USB sound card, relaying audio feedback clearly to the user. The system is powered by a 24,000mAh power bank. A USB-C PD to 15V 5A DC converter is sufficient to ensure the Jetson Orin Nano receives the necessary power for continuous operation.

The harness itself underwent multiple variations until we arrived at a version that looked aesthetic, was comfortable to wear for people of all builds, and adequately protected the components from the environment.

6.2 Walk Sign Classification Subsystem

The walk sign classification subsystem is responsible for detecting the “WALK” and “STOP” signals in real-time. It utilizes a CNN-based ResNet classifier [6] trained to achieve an AUROC of ≥ 0.9 on a test set, which should high reliability in real-world conditions. Additionally, we enhance model robustness majority voting, where consecutive frame predictions are majority-voted over. Since at least 90% of individual frames must be correctly classified, this reduces the probability of five consecutive misclassifications to less than 1%, increasing user safety and trust in

the system. Refer to Figure 3 for a flow chart detailing the image collection and model output.

To meet real-time constraints, the model inference latency must be ≤ 3 seconds per frame. This ensures that users receive timely "WALK" or "STOP" signals without unnecessary delays. The classification results are continuously monitored, and once five consecutive frames indicate a "WALK" signal, the system transitions to the navigation subsystem, informed by the obstacle detection subsystem. This is to transition from waiting at the crosswalk to crossing the road.
Originally, the ResNet model was trained using Python's tensorflow libraries in Google Colab, and the resulting .h5 model weights were loaded onto the Jetson Orin Nano. However, during integration, we found that the tensorflow version required was incompatible with the Docker container required to run models using the GPU. The solution was to rebuild the same ResNet model using Python's torch library instead, completely avoiding tensorflow.

6.3 Obstacle Detection Subsystem

The obstacle detection subsystem is designed to identify unexpected objects in the user's path using a YOLOv12-based object detection model. To ensure robust decision-making, the system follows the same majority voting strategy as the walk sign classifier. The model must correctly classify at least 90% of individual frames, ensuring the probability of three out of five consecutive frames being misclassified remains below 1%. This helps prevent critical false negatives, enhancing overall safety. Refer to Figure 4 for a flow chart detailing the image collection and model output. Unlike the navigation phase, this subsystem works in parallel with the crosswalk navigation phase, which is reflected in the flow chart.

Real-time constraints dictate that obstacle detection must provide feedback within 1.5 seconds, with an inference latency of ≤ 1 second per frame, which is lower than the walk sign classification. However, this more stringent inference speed requirement allows users sufficient reaction time to navigate around detected obstacles safely.

One issue that arose when integrating the YOLO model with the Jetson was that the inference latency was very high, taking almost 10 seconds to output a single prediction. We discovered that the program was not detecting the Jetson's GPU, so a Docker container was required to run the YOLO model, reducing latency to less than 100 ms per prediction.

6.4 Crosswalk Navigation Subsystem

The crosswalk navigation subsystem ensures that users maintain a safe and effective walking trajectory. It employs data from our ML model to detect deviations from the intended path, with a tolerance of $\pm 45°$. The system maintains an accuracy of at least 90% in detecting deviations, ensuring reliable path guidance. In addition, it continuously monitors the surrounding via the image input from the camera, and appropriately alerts the user to take any necessary deviations to avoid potential obstacles if the YOLO model identifies nearby obstacles (such as right in front of the user) with a high degree of confidence. When a user veers by more than 45° without any obstacles detected, corrective audio feedback is generated within 1 second to prompt realignment. The system leverages the pyttsx3 text-to-speech engine to provide clear and immediate auditory cues. Refer to Figure 4 to see how this subsystem interacts with the obstacle detection subsystem.

Originally, we had planned on using a GPS module to also pinpoint the position of the traffic sign to aid in calculating the deviation. However, we found that just using the IMU sensor was sufficient.

6.5 Audio Feedback Management

To ensure seamless user experience, all cues from the Walk Sign Classification, Obstacle Detection, and Crosswalk Navigation subsystems are managed through an audio feedback system. This system prioritizes the most critical alerts to prevent information overload and confusion. The audio manager selects between text-to-speech notifications and pre-recorded audio alerts, ensuring that users receive clear and immediate guidance. The feedback system's response time is optimized to ≤ 1 second from event detection to cue playback. User studies were conducted to refine audio cues and ensure that they effectively communicate necessary guidance without causing cognitive overload.

One issue that arose when adding the audio components was that despite unit tests succeeding on the board, once we moved the logic to the program running in the Docker container, the audio stopped working. This required some debugging of the Docker container's permissions until finally the audio worked with the rest of the system.

7 TEST & VALIDATION

7.1 Walk Sign Classification Subsystem

To evaluate the walk sign classification subsystem, we used our test set that contains images under various lighting and weather conditions, as well as test our models in a real world setting in a controlled crossing scenario. The system's performance was assessed using AUROC for walk signal classification. The classification model must achieve an AUROC greater than 0.9 on our test set to be considered reliable. Additionally, majority vote classification accuracy were assessed using 5-frame sequences from the test dataset. The accuracy was calculated based on the proportion of sequences where the majority prediction is

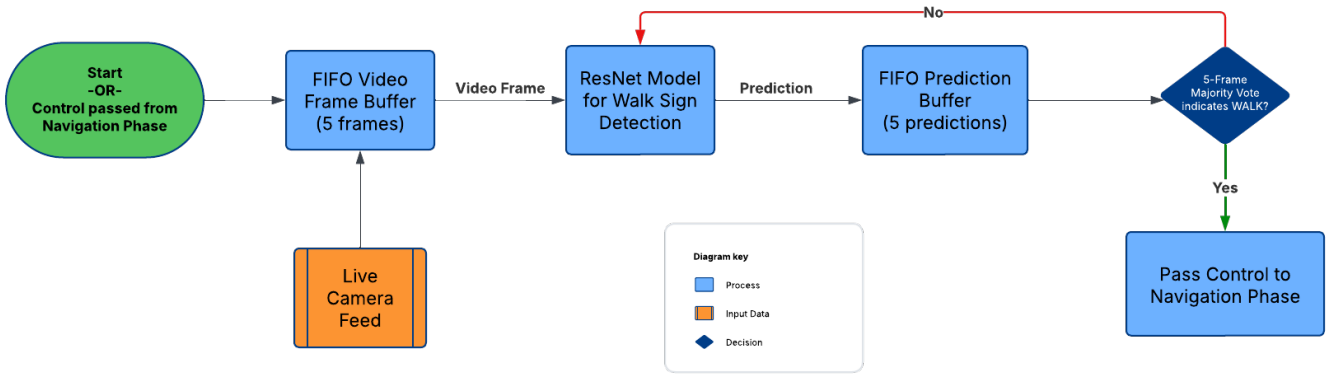

Figure 3: The flow chart for the walk sign classification subsystem.

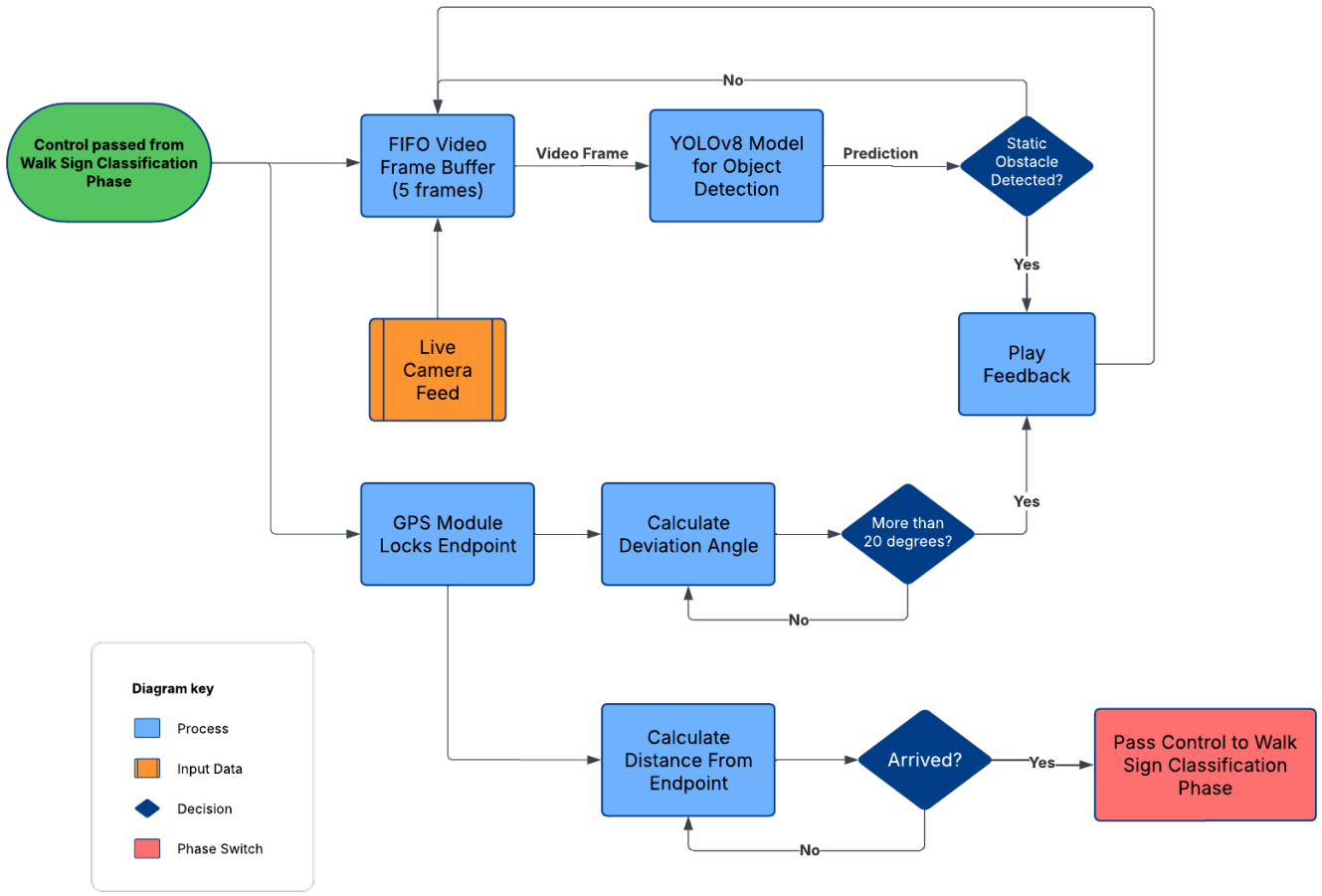

Figure 4: The flow chart for navigation subsystem, which combines the obstacle detection phase and crosswalk navigation phase.

correct. A minimum accuracy of 95% is required across test sequences. Inference efficiency was also measured by evaluating the processing time per frame and batch. Each frame should be processed within 100 milliseconds, and the total audio delay must not exceed 500 milliseconds. Refer to Figure 8 for the full table of results. Here is also a confusion matrix from testing the image classification on a test dataset:

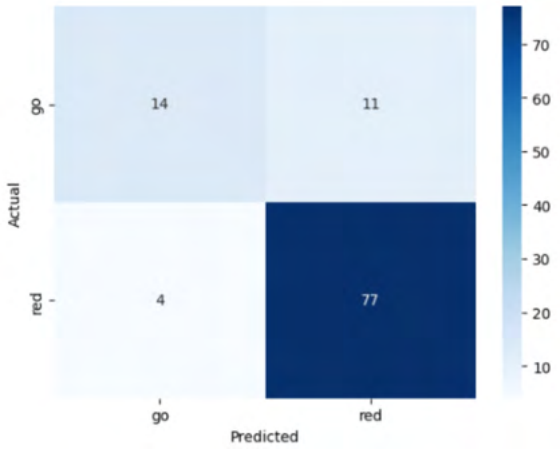

Figure 5: The results of testing the ResNet model on images collected around Pittsburgh. We see that the model rarely predicts false positives, and that the lower test accuracy is mostly due to false negatives, which are much safer and more tolerable for our use case.

7.2 Obstacle Detection Subsystem

To evaluate the obstacle detection subsystem, we once again sourced real world images under various lighting and weather conditions, as well as tested our models in a real world setting in a controlled crossing scenario. The system's performance was assessed using mean Average Precision (mAP) for object detection, and we used the bdd100K dataset to initially proxy our models' performance on an out of distribution dataset that the models would not have been trained on, and therefore would not have seen before. This dataset contains over 100,000 images collected around the world in varying lighting conditions. The YOLO detection models were originally expected to achieve an mAP of >= 0.7 on all classes. However, we found that none of the models that we tried (YOLOv12 10 and RT-DETR (11, among others) were not able to meet these initial guidelines. However, we observed that all of our tested models topped out at around the same values, and hypothesized that this was simply the best that object detection models could do on our out of distribution evaluation set. When evaluating the object detection models on the real crosswalk images that we manually corrected, we found that the object detection models performed extremely well, correctly locating and identifying objects in the camera's field of view very consistently, referring to ẽffig:frame1 and 13. We therefore posit that despite not meeting our original evaluation requirement, the object detection models perform sufficiently well in our real world setting we are deploying in.

To verify the obstacle detection and classification subsystems, user validation tests were conducted where participants wear the system and attempt to detect obstacles in their path. The system's accuracy was evaluated based on the percentage of users successfully orienting the camera to detect signals within five seconds. This helped us make improvements to the mounting instructions, and add auditory guidance to assist in proper positioning. Refer to Figure 8 for the full table of results.

7.3 Crosswalk Navigation Subsystem

The crosswalk navigation subsystem was assessed using controlled veering angle detection tests. Participants walk along a straight path and at predefined angles of 10°, 30°, 45°, and 60° while the system detects deviations. The system must correctly identify at least 95% of deviations of 45° or greater. Additionally, the response time for veering detection was measured by recording the delay from detection to auditory feedback. The response time must not exceed 300 milliseconds. Refer to Figure 9 for the full table of results.

7.4 Power Draw

The overall power efficiency of the crosswalk navigation subsystem was tested by measuring power consumption under peak load conditions. The system must operate for at least six hours without requiring a recharge. Refer to Figure 9 for the full table of results.

7.5 Audio Feedback Management

To assess the clarity and effectiveness of audio feedback, users intentionally veered off-course and provide subjective ratings on a 1–5 scale. The system must achieve an average clarity score of at least 4.0. In addition to clarity, the system's response time for providing audio feedback was evaluated. The delay from event detection to auditory response must not exceed 300 milliseconds
By conducting these evaluations, the smart chest harness's reliability, efficiency, and usability were verified, ensuring it effectively assists visually impaired users in crosswalk navigation and obstacle detection. Refer to Figure 9 for the full table of results.

8 PROJECT MANAGEMENT

8.1 Schedule

The schedule is shown in Fig. 7. The only changes to the Gantt chart since the design review report were the red blocks. This was mostly debugging integration issues such as Python library dependency conflicts, getting the audio to work, and setting up the Docker container. There was also extra time added for optimizing the two computer vision models and improving their performance.

8.2 Team Member Responsibilities

Our work and responsibilities was distributed to team members as follows.

- **William:** Primarily responsible for the Hardware. This includes: Ordering parts, testing parts, setting up devices; interfacing components, assembly and physical design.
- **Max:** Primarily responsible for the Walk Sign Classifier Subsystem. This includes optimizing/training the Walk Sign Classifier, writing the control system, and testing the model.
- **Andrew:** Primarily responsible for: Navigation Subsystem; This includes optimizing/training the Object Detection model, writing the control system, and testing the model.
- **Shared:** The shared tasks include integrating between the hardware and software interface for components, testing for requirements, user testing, and making final optimizations.

8.3 Bill of Materials and Budget

Refer to Table 1. The only item that we did not use was the GPS module. We did not order any additional materials, though free materials were used from IDeATe courses and facilities.

8.4 TechSpark Usage Plans

We did not use TechSpark for our project. Instead, the extra free materials (such as acrylic), lasercutting, and assembly were done through IDeATe.

8.5 Risk Mitigation Plans

There are a few potential failure points in our project, outlined below with their respective mitigation plans:

1. **Risk:** Walk Sign Classifier's accuracy may be reduced in poor visibility conditions.
   Mitigation Plan: We retrained with nighttime data, which did improve its test accuracy. Another solution that we considered was using a different, pretrained model besides the ResNet one we used.

2. **Risk:** A big concern is that the inference speed of our models may be too high. This would cause issues like audio cues being given long after the event has occurred, potentially placing the user in data.
   Mitigation Plan: In this case, we planned to optimize the model through quantization. This is a technique that allows us to reduce the size and computational requirements of the model, without sacrificing too much accuracy. However, we found that using a Docker container to ensure that the models used the NVIDIA GPU was another way to reduce the latency.

3. **Risk:** Camera placement and orientation greatly affects the output quality of our ML models. If a user improperly wears the system, the video data may be poor/unusable.
   Mitigation Plan: We planned to make the system as fool-proof as possible to wear, so that the relative position of the camera on a user's body would be constant. We mitigated this risk by using adjustable components anytime there was a contact point between the user and the project (e.g. chest harness with adjustable strap). The mounting mechanism was also designed to be snug and secure to prevent excessive wobbling. User testing with people of different builds showed that our design succeeded in this case.

4. **Risk:** The project could potentially fail to detect the user veering off the course of the sidewalk. The IMU sensor output could potentially drift, causing incorrect values to be read.
   Mitigation Plan: There are a few ways we could have mitigated this. First, we could continuously re-estimate the target position as the user crosses the road, instead of just at the beginning. This would correct for drift error, and also ensures that we do not rely on a single estimated target endpoint. We could have also tried to integrate the GPS module, and use that data to track the user's location on the sidewalk. However the GPS solution came with its own set of risks, relating to the accuracy of the module and reliance on the OpenStreetMap or Google API. In the end, the IMU sensor was sufficient by itself and the drift did not prove to be consequential.

5. **Risk:** For the audio output, there could potentially be delayed feedback depending on the latency of the Audio Manager process and protocol used (i.e. Bluetooth).
   Mitigation Plan: Latency issues stemming from the Audio Manager process were likely resolvable through code optimizations, as we do not envision that being a particularly resource intensive process. However, if the audio latency issues came from the Bluetooth protocol (which is used by the bone-conducting headphones), we planned to swap to using the on-ear headphones that use the 3.5mm audio jack. This would remove the latency caused by a wireless protocol, as wired audio have near zero latency (for such a short wire length).

6. **Risk:** A concern is that the power consumption of the system might be too high. We calculated for the reported 15W upper limit of the Jetson Orin Nano, but we may need to use a "Super Mode" which consumes even more power in order to reduce the inference latency of our models.
   Mitigation Plan: The simplest solution was to just buy a larger power bank (24,000mAh). However this goes to our considerations for user comfort, as a power

Table 1: Bill of materials

Description	Model #	Manufacturer	Quantity	Cost @	Total
Processing Unit	Jetson Orin Nano	Nvidia	1	$249 ($0 In Stock)	$0
Camera	IMX219	Arducam	1	$23.99	$23.99
GPS	PA1616S	Adafruit	1	$29.95	$29.95
9-DOF IMU	BNO055	Adafruit	1	$50.18	$50.18
USB-C PD to DC Cable	5451	Adafruit	1	$7.95	$7.95
Headphones	OpenMove	AfterShokz	1	$129.99	$129.99
M2.5 Nylon Standoffs	pta241226tt001308	PATIKIL	1	$13.49	$13.49
M2 Nylon Standoffs	B0BXT4FG1T	COMRUN	1	$12.99	$12.99
					$385.13

bank of that size is quite heavy. As such, we preferred to mitigate this risk through other means like optimizing for power management and adding low-power states. In the end, the 24,000 mAh battery we used provided enough battery to meet our requirements.

9 Ethical Issues

The Self-Driving Human project has several important ethical considerations related to the safety, autonomy, and well-being of its users; visually impaired pedestrians. While the goal of the system is to increase independence and safety in urban navigation, we also need to evaluate the ethical implications of its potential deployment.

The first ethical priority is user safety. A failure in the system's ability to correctly identify crosswalks, interpret traffic signals, or detect hazards such as oncoming vehicles, construction, or uneven terrain could lead to serious harm. This is especially critical because users may place a high degree of trust in the system. To mitigate these risks, extensive validation through simulation and controlled real-world testing is essential before public deployment, which we have begun to implement in our preparation for our demo. Backup safety mechanisms such as haptic feedback or emergency alerts could also be included to reduce the risk of harm in the event of a malfunction. Additionally, the system should be designed to complement, not replace, existing mobility aids like white canes or guide dogs to avoid promoting over-reliance on the technology.

Next, there is a risk of exposing sensitive user information such as location, movement patterns, or video recordings. Ensuring strong encryption, anonymization, and local processing where possible is critical to upholding user privacy. Ethical development should prioritize user consent, transparency in data usage, and compliance with global data protection regulations (one example is GDPR).

The Self-Driving Human also has the potential to enhance public welfare by expanding opportunities for visually impaired individuals to live independently, access education, employment, and participate more fully in society. However, economic barriers could limit access to the technology, especially for low-income users or in under-resourced regions. Ethical design should include strategies to ensure affordability, such as public funding, nonprofit partnerships, or insurance coverage. Failure to address these disparities could reinforce existing inequalities.

Additionally, navigation environments vary significantly around the world in terms of infrastructure, signage, cultural behaviors in traffic, and urban planning. A system trained solely on data from one country (such as the U.S.) may perform poorly in international settings, potentially endangering users abroad. Therefore, global dataset diversity must be prioritized during model training. Moreover, collaboration with local communities and disability organizations worldwide is essential to tailor the device to diverse user needs and environments.

Finally, the environmental impact of hardware production and energy consumption must be considered. The system should use sustainable materials where possible and be designed for energy efficiency and long-term durability. Ethical engineering also includes planning for responsible e-waste disposal and repairability. Other measures we have preemptively took to address this is to use compressed model inference tehcnique such as quantization to reduce the computational costs of continuously running model inference.

The success of the Self-Driving Human depends not only on technical performance but also on how well it addresses the ethical challenges associated with safety, equity, privacy, and sustainability. Incorporating these ethical principles into the design and implementation process is essential to creating a product that serves the public good and earns the trust of its users and society at large.

10 RELATED WORK

Yu et al [7] implemented a lightweight solution to the navigation problem, using CNNs to provide direction and information on the pedestrian crossing light, which is available through an iOS app. However, this approach does not take into account any sort of object detection, which

poses a significant risk to the user in our project definition. Cai et al. [1] offers an impressive analysis of several ML paradigms to determine the best models for ensuring pedestrian safety, but do not incorporate any hardware integration in their solution, which makes it inapplicable for our project scope. We noticed that Hua et al. [3] proposed a multimodal ML model, using information from both regular and infrared images to guide a more accurate analysis of pedestrian crossing obstacles. However, this work does not incorporate any navigation for the user, making this incomplete with regards to our application. Finally, Hwang et al. [4] developed a custom multimodal Vision-Language Transformer Model to provide both a safety score and a description of the street in natural language, providing a nuanced understanding of the surroundings for the user. This is an exciting new approach to the problem at hand, but they do not explicitly use it for any use case beyond determining when it is safe to cross the road. Arguably, our current control modules integrating both the walk sign classifier and the navigation achieve a similar goal, and so this work does not provide much benefit beyond using a different approach to solve the same problem.

11 SUMMARY

The Self-Driving Human is a navigation aid designed to help visually impaired pedestrians safely cross the street. It does so using computer vision, IMU-based heading tracking, and real-time audio feedback. The device provides users with walk sign assistance, obstacle avoidance, and cross walk alignment aid. We envision our project providing visually impaired users with a greater level of independence by removing their reliance on external accessibility infrastructure, offering navigation support in its place.

Overall, we were able to meet most of our design requirements and integrate all submodules into one complete system. The only metric we did not meet was the mAP of the off-the-shelf YOLOv12 model, but we qualitatively showed that its performance in detecting objects relevant to our use case was high enough.
Regarding future work, we would like to improve upon one of the fundamental limitations of our project, which was the latency of our audio feedback, which was a result of model inference and text-to-speech delay. Although we met our required latency thresholds, faster feedback could result in an even smoother experience for the users. One solution could be to avoid using text-to-speech libraries and instead simply play pre-recorded audio files, avoiding the processing delay of generating audio from text.

One lesson we learned was to begin the complete integration process earlier, as submodules that work independently may not always work together. This is true for both hardware and software components, and in our case, the integration of the ResNet model, YOLO model, and audio functions were the main problems. Python dependency conflicts can be incredibly annoying to resolve, and a good rule of thumb when sourcing code is to use the most up-to-date versions you can find. Another lesson is to set up a Github repository at the very beginning of the project, as organization becomes harder to manage if you start it later.

Glossary of Acronyms

- AI – Artificial Intelligence
- AUROC – Area Under the Receiver Operating Characteristic Curve
- CNN – Convolutional Neural Network
- DC – Direct Current
- FPS – Frames Per Second
- FOV – Field of View
- GPS – Global Positioning System
- IMU – Inertial Measurement Unit
- mAP – Mean Average Precision
- ML – Machine Learning
- PD – Power Delivery
- ReLU – Rectified Linear Unit
- ResNet – Residual Neural Network
- SDK – Software Development Kit
- TTS – Text-to-Speech (used for converting navigation instructions into voice feedback)
- USB – Universal Serial Bus
- YOLO – You Only Look Once

References

[1] Jun Cai, Mengjia Wang, and Yishuang Wu. "Research on Pedestrian Crossing Decision Models and Predictions Based on Machine Learning". In: *Sensors* 24.1 (2024). ISSN: 1424-8220. DOI: 10.3390/s24010258. URL: https://www.mdpi.com/1424-8220/24/1/258.

[2] Kaiming He et al. "Deep Residual Learning for Image Recognition". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2016, 1, p. 1. DOI: 10.1109/CVPR.2016.90. arXiv: 1512.03385 [cs.CV].

[3] ChunJian Hua et al. "Pedestrian detection network with multi-modal cross-guided learning". In: *Digital Signal Processing* 122 (2022), p. 103370. ISSN: 1051-2004. DOI: https://doi.org/10.1016/j.dsp.2021.103370. URL: https://www.sciencedirect.com/science/article/pii/S1051200421004097.

[4] Hochul Hwang et al. "Is it safe to cross? Interpretable Risk Assessment with GPT-4V for Safety-Aware Street Crossing". In: *arXiv e-prints*, arXiv:2402.06794 (Feb. 2024), arXiv:2402.06794. DOI: 10.48550/arXiv.2402.06794. arXiv: 2402.06794 [cs.CV].

[5] Glenn Jocher, Ayush Chaurasia, and Jing Qiu. *Ultralytics YOLO*. Jan. 2023. URL: https://ultralytics.com.

[6] Dense Lance. *resnet-simple*. 2021. URL: https://github.com/DenseLance/resnet-simple%7D,note={Version1.0,Computersoftware}.

[7] Samuel Yu, Heon Lee, and Jung Hoon Kim. "Street Crossing Aid Using Light-weight CNNs for the Visually Impaired". In: *arXiv e-prints*, arXiv:1909.09598 (Sept. 2019), arXiv:1909.09598. DOI: 10.48550/arXiv.1909.09598. arXiv: 1909.09598 [cs.CV].

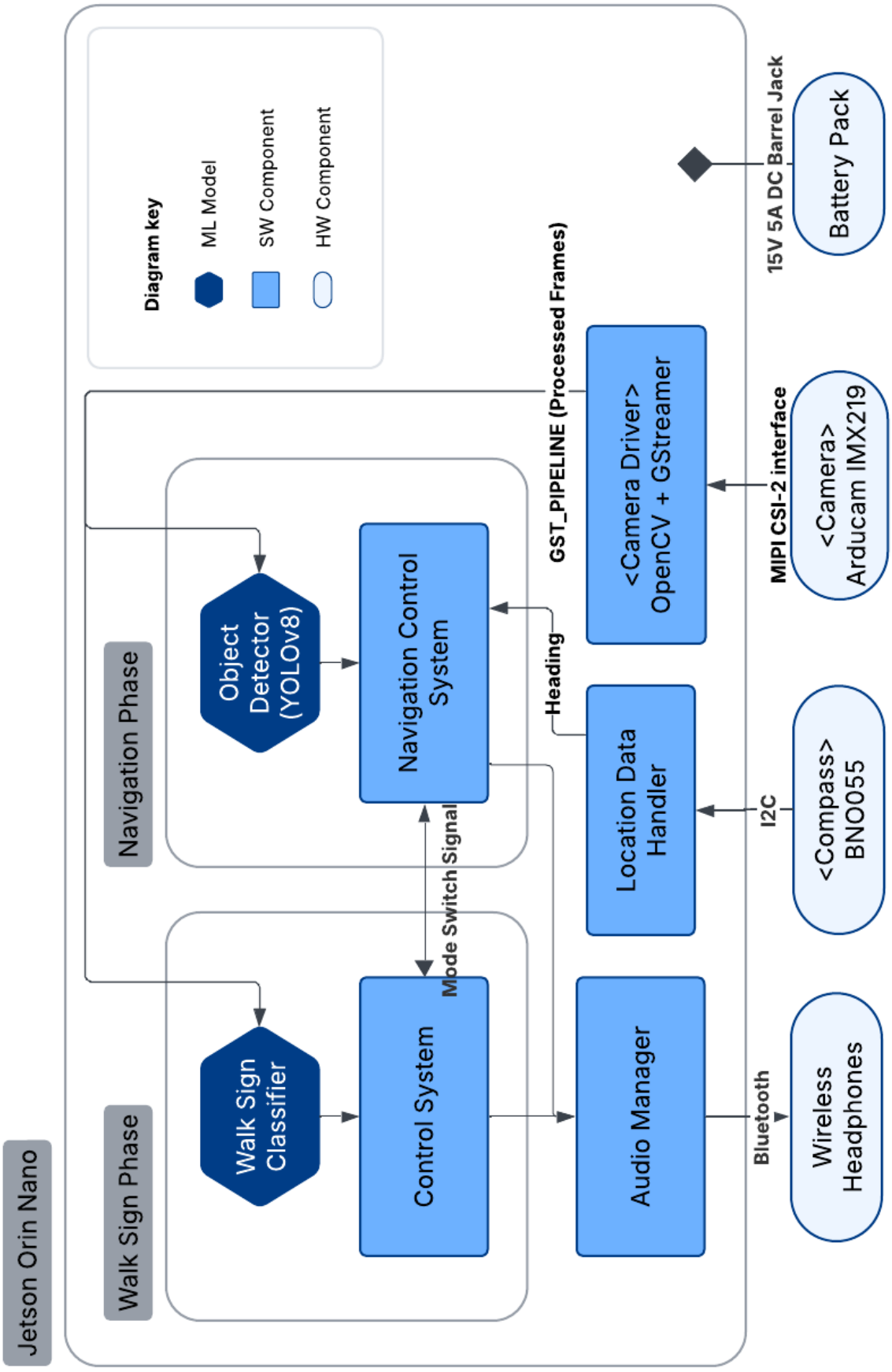

Figure 6: A full-page version of the same system block diagram as depicted earlier.

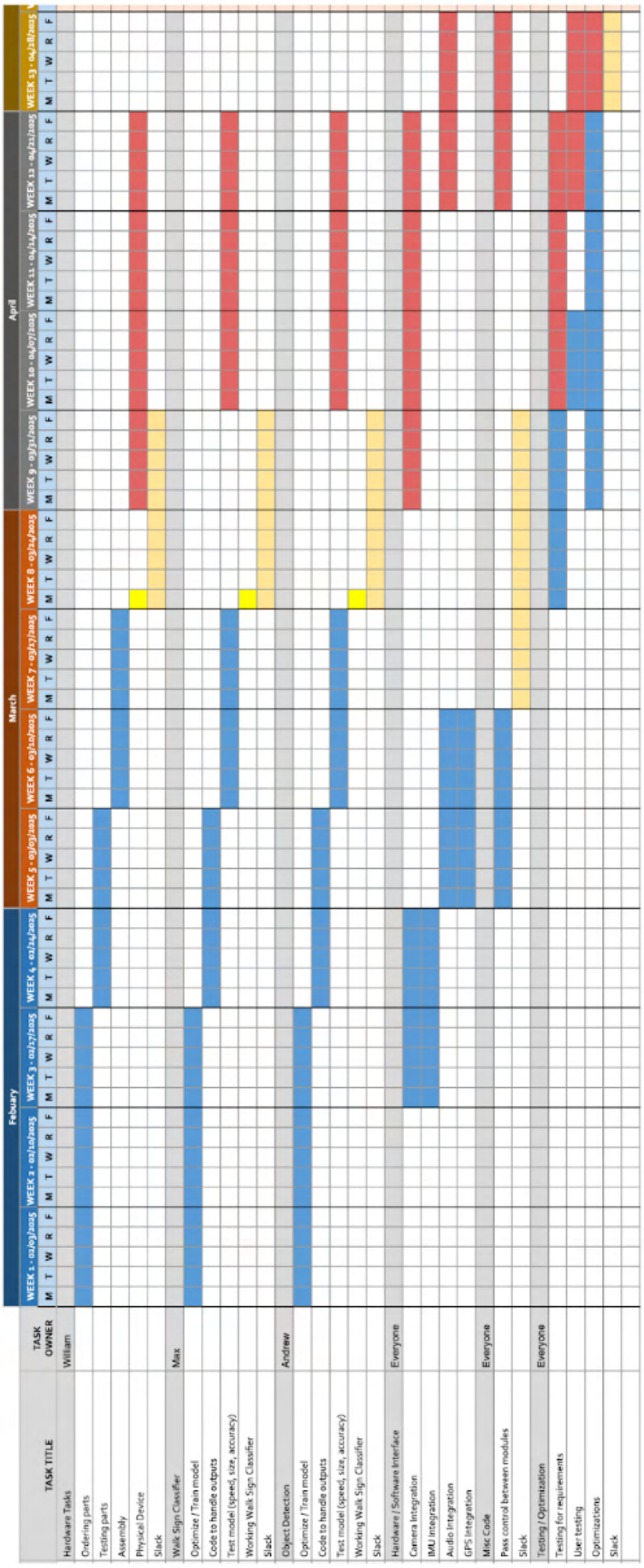

Figure 7: Gantt Chart

Test Name	Test Inputs	Test Outputs	Passing Criteria	**Results**
Per-frame Classification Performance	Real-world video frames from crosswalks under various lighting and weather conditions.	AUROC (Walk SIgn Classification) mAP (Object Detection)	AUROC > 0.90 accuracy > 85% mAP > 0.7	Walk Sign Model: - AUROC: 0.936 - accuracy: 87.2% Object Detection Model: - mAP: 0.3-0.6
Majority Vote Classification Accuracy	5-frame sequences from test dataset.	Accuracy: Correct majority vote predictions/Total number of sequences	≥ 95% accuracy across test sequences.	5-frame majority vote accuracy: 98.2%
Inference & Audio Response Time	5-frame inference batches	Inference time per frame and batch; audio delay.	Inference latency ≤ 100ms/frame; audio delay ≤ 500ms.	Walk Sign Latency: 33 ms Object Detection Latency: 90 ms Audio Latency: 100 ms
User Verification	10 users of different builds wear device and attempt to detect walk signals and obstacles 5 times each.	% of users successfully orienting camera to detect signals.	≥ 90% success rate within 5s.	Successful orientation in 95% of cases

Figure 8: The results of testing our computer vision models, feedback latency, and user verification testing.

Test Name	Test Inputs	Test Outputs	Passing Criteria	**Results**
Veering Angle Detection	Controlled user tests walking along a straight path and at predefined angles (10°, 20°, 30°).	Correctly detecting 20°+ deviations/All 20°+ deviations	≥ 95% accuracy for 20°+ deviations.	To Be Determined (drift correction TBD)
Audio Response Time	Users simulate veering (>20°); measure delay from detection to audio.	Mean delay (ms).	≤ 300ms response time.	≤ 100ms response time.
Audio Feedback Clarity	Users intentionally veer off-course and provide subjective ratings.	Clarity score (1–5 scale).	Mean clarity rating ≥ 4.0	4.0 average score
Power Draw Test	Measure device power consumption under maximum load.	Battery life (hours) at peak wattage.	≥ 6 hours of operation.	Idle (4.1W) WalkSignClassifier (7.1W) ObjectDetection (17.0W) Max: 8 hrs Min: 3 hrs

Figure 9: The results of testing the angle detection, audio performance, and the power draw.

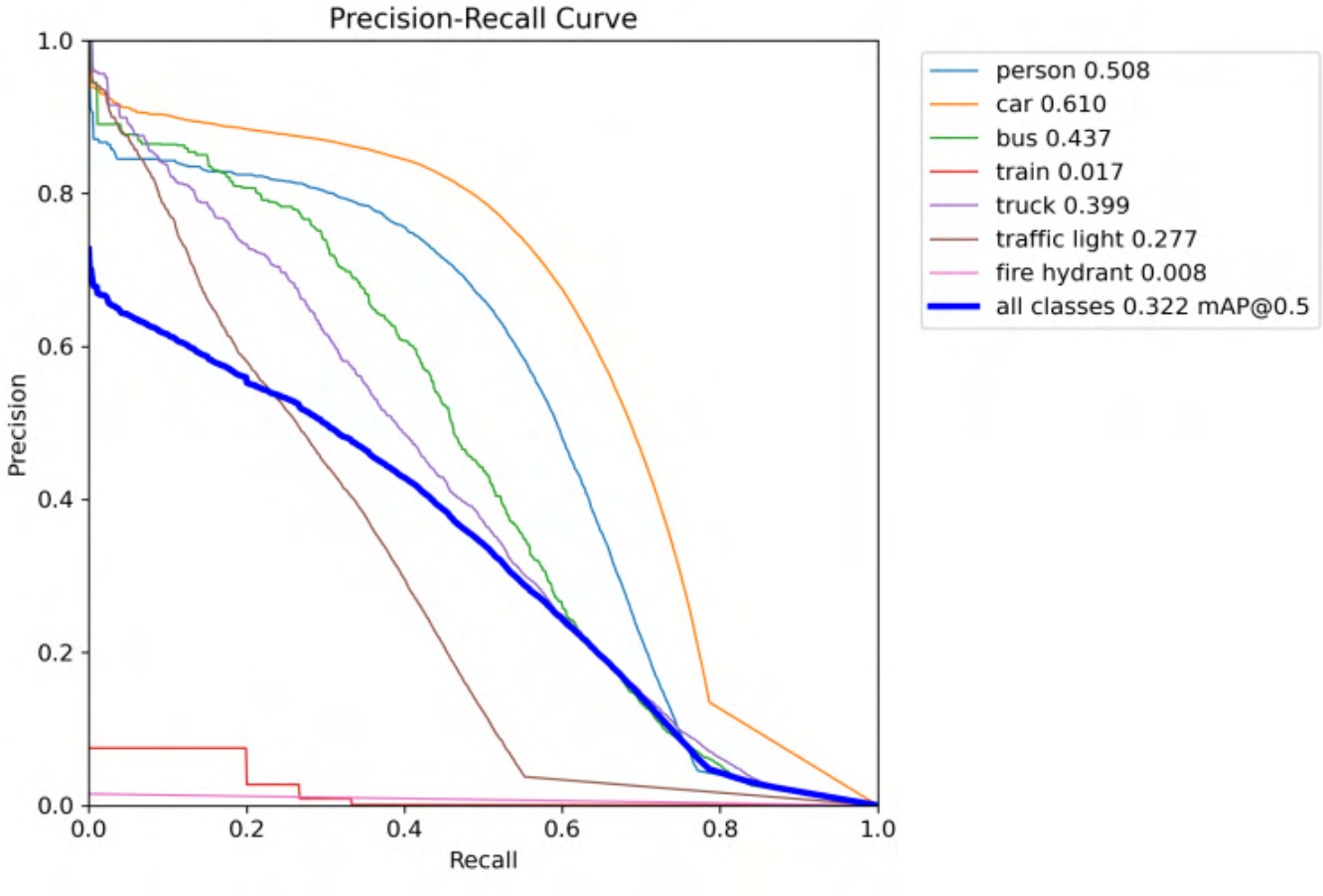

Figure 10: Precision-recall curve of the YOLOv12 object detection model, plotted for all classes with the mAP displayed in the legend.

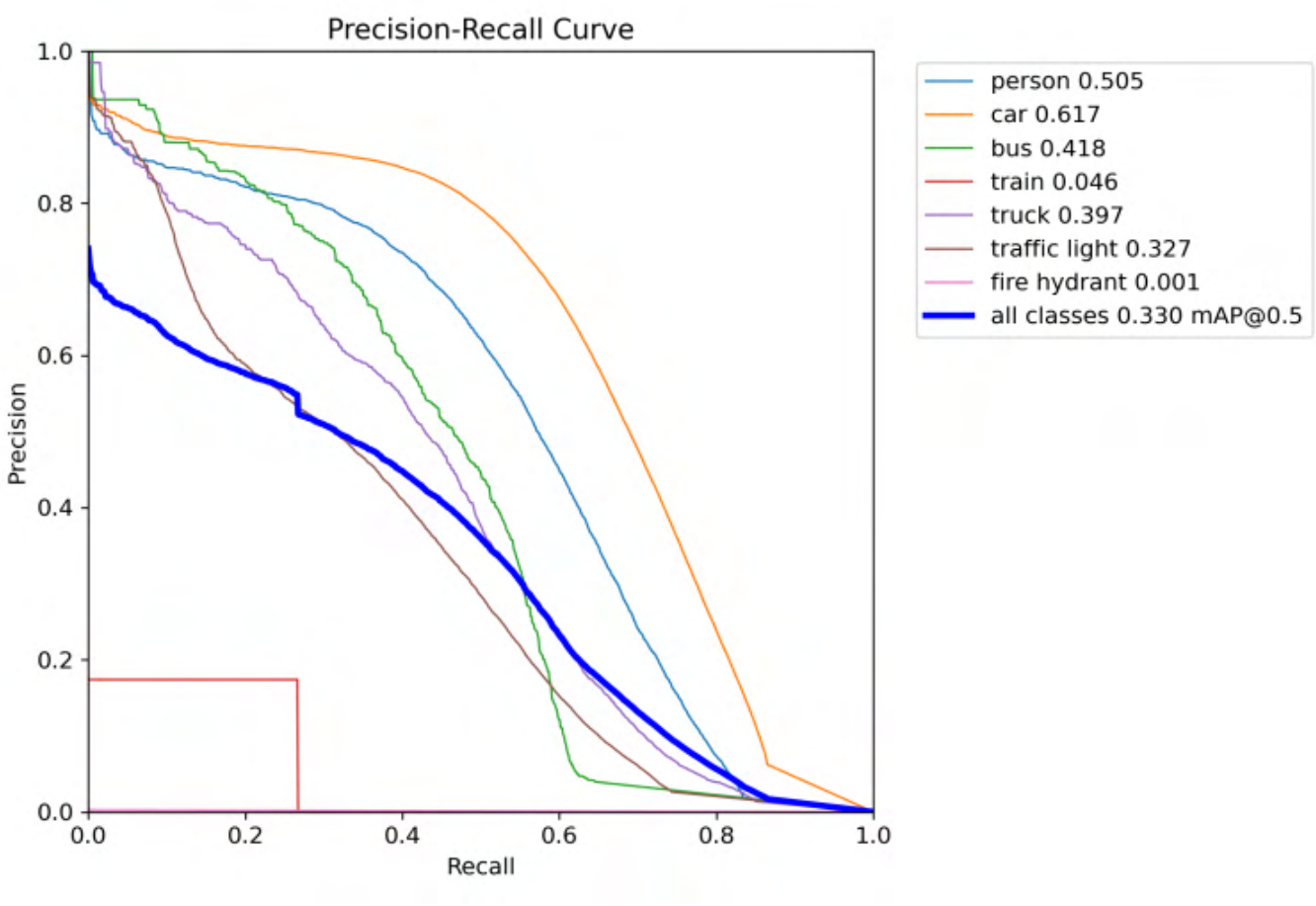

Figure 11: Precision-recall curve of the YOLOv12 object detection model, plotted for all classes with the mAP displayed in the legend.

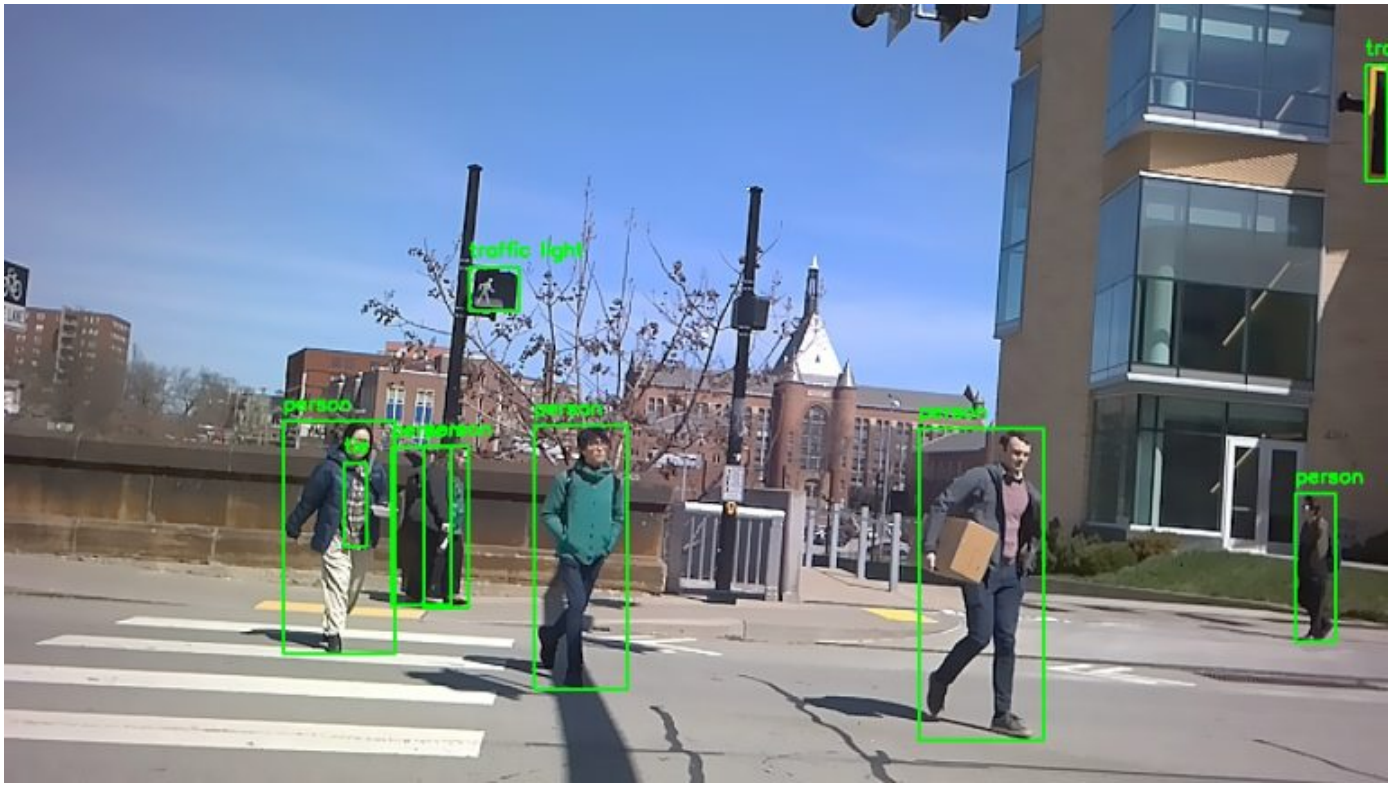

Figure 12: Bounding boxes on an input image that was used during testing. We observe that not only are the locations correct, but the classifications of of the objects are completely correct.

Figure 13: Another example of our object detection model output on our real world dataset that we curated. Once again, we observe that not only are the locations correct, but the classifications of of the objects are completely correct.