Methodology

Human Detection and Categorization Module

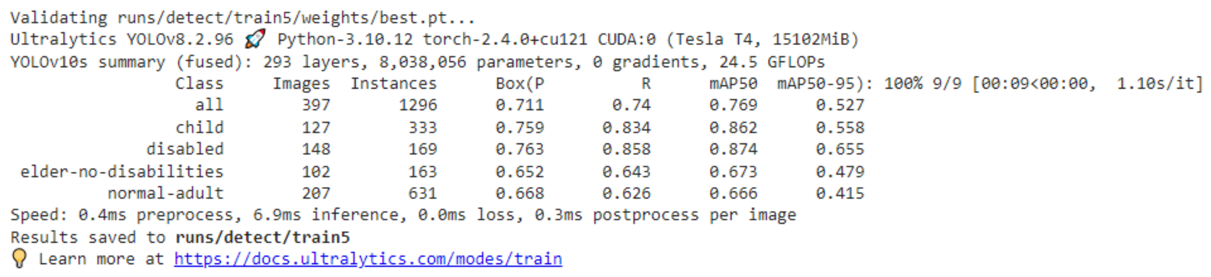
We developed a custom dataset containing images of human agents in various environments to train a YOLOv10-small model [1] for detecting and categorizing human agents. This dataset included images derived from the study on Vulnerable Human Agent Detection [2], which explored pedestrian categorization to enhance autonomous vehicle navigation. The dataset primarily focused on four human agent classes, categorized based on their physical appearances: child, normal adult, elder (non-disabled), and disabled.

To improve the dataset's diversity and robustness, we applied data augmentation techniques such as rotation, contrast adjustment, brightness variation, noise injection, and flipping. These methods expanded the dataset's coverage of potential light conditions and addressed challenges arising from the robot's actual camera input. The final augmented dataset consisted of 4,796 images distributed as follows:

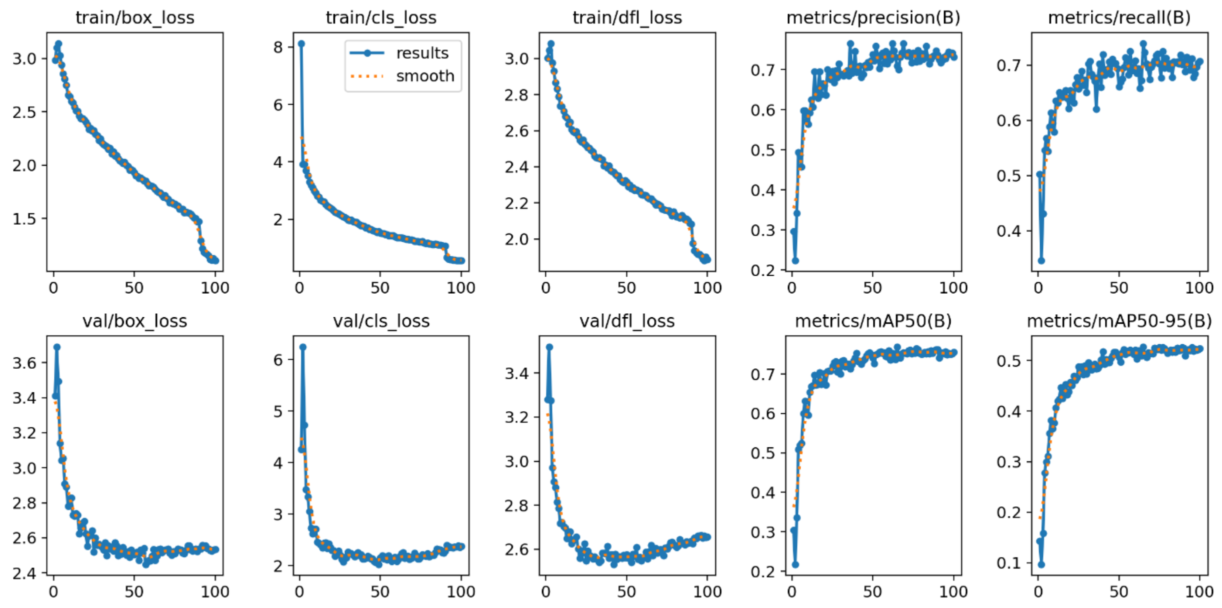
- Training set: 4,003 images
- Test set: 397 images
- Validation set: 196 images

We ensured that test and validation sets were not augmented and did not overlap with the training set, thus maintaining the integrity of performance evaluations. We trained the YOLOv10-small model using this dataset, achieving notable performance metrics.

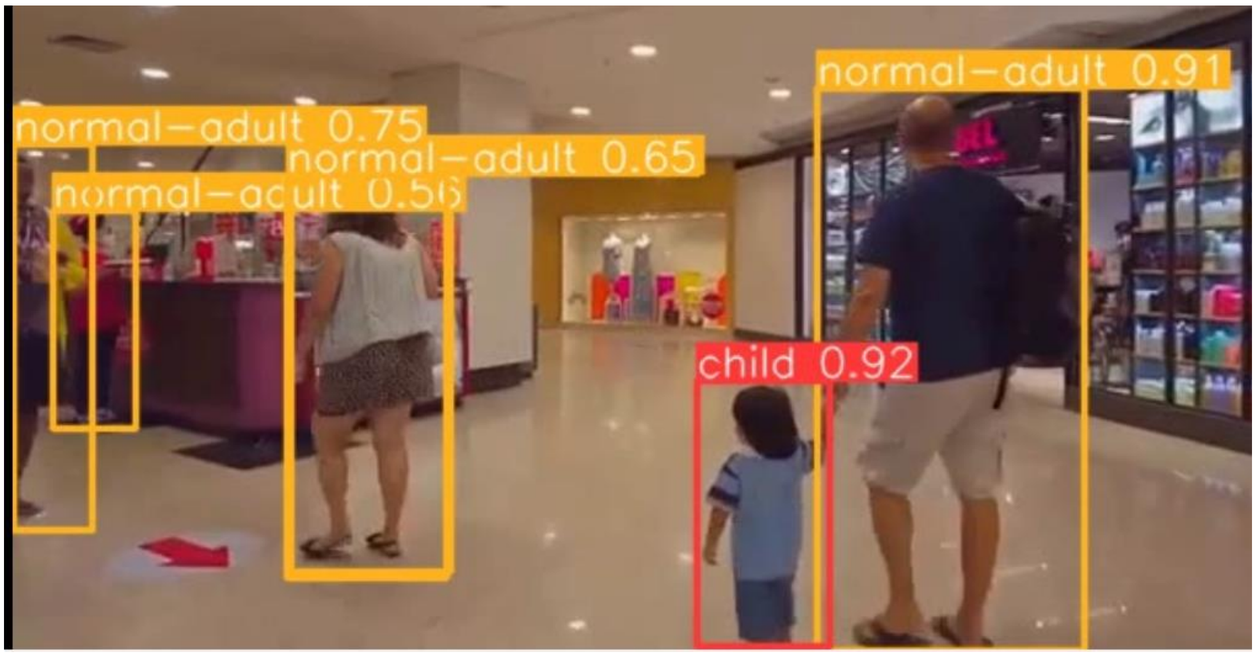
The model performed optimally at epoch 61, which corresponded to the point of optimal validation performance. Using early stopping, we terminated training at this point and saved the best-performing model weights. Validation of the model on the test set demonstrated reasonable precision and recall metrics for each class, as illustrated in the following results.

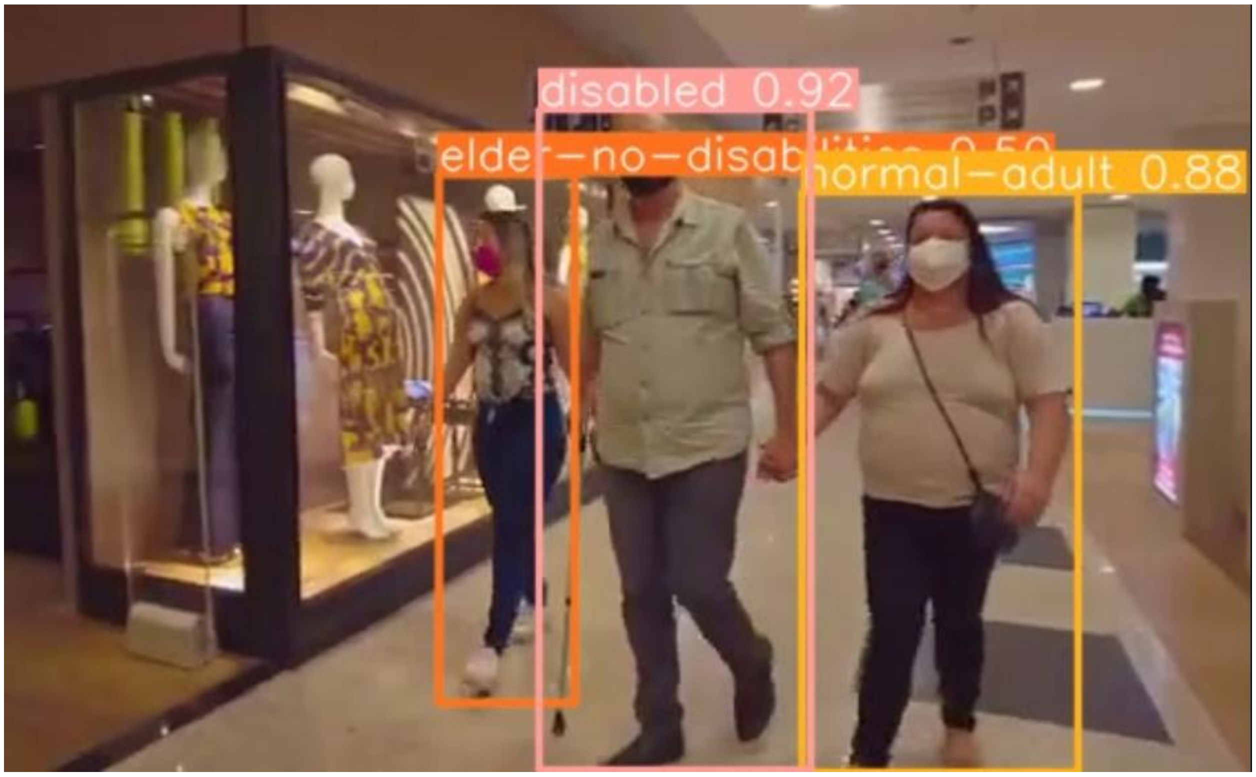


```
Validating runs/detect/train5/weights/best.pt...
Ultralytics YOLOv8.2.96 🚀 Python-3.10.12 torch-2.4.0+cu121 CUDA:0 (Tesla T4, 15102MiB)
YOLOv10s summary (fused): 293 layers, 8,038,056 parameters, 0 gradients, 24.5 GFLOPs
                Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 9/9 [00:09<00:00,  1.10s/it]
                  all        397       1296      0.711       0.74      0.769      0.527
                child        127        333      0.759      0.834      0.862      0.558
             disabled        148        169      0.763      0.858      0.874      0.655
 elder-no-disabilities        102        163      0.652      0.643      0.673      0.479
         normal-adult        207        631      0.668      0.626      0.666      0.415
Speed: 0.4ms preprocess, 6.9ms inference, 0.0ms loss, 0.3ms postprocess per image
Results saved to runs/detect/train5
💡 Learn more at https://docs.ultralytics.com/modes/train
```



After detecting human agents and their classes in each video frame, the model outputs bounding box coordinates along with the corresponding class of each agent. We then transform these 2D coordinates into a 3D reference frame and cross-reference them with those identified through LiDAR data. This process uniquely identifies human agents and assigns their classes within the dynamic environment map.





The acquired class data provides insights into each human agent’s movement capabilities and related factors. We use this information for human path planning, which feeds into the robot’s path planning process. Additionally, the class data defines the human “footprint” in the 2D environmental map. Instead of representing human agents as dots, we use circles to represent their footprints. These footprints account not only for the physical area occupied by each human agent but also for the uncertainty in their movements, which varies depending on their class and other factors. For instance, children tend to exhibit more unpredictable movements than adults, resulting in larger footprints to reflect this higher uncertainty. By incorporating these class-dependent footprints into the 2D map, the robot can better predict and adapt to dynamic human behavior. This enables the robot to plan its path more effectively, navigating crowded environments while maintaining appropriate distances and ensuring socially acceptable interactions.

Power Management System and Self-Charging

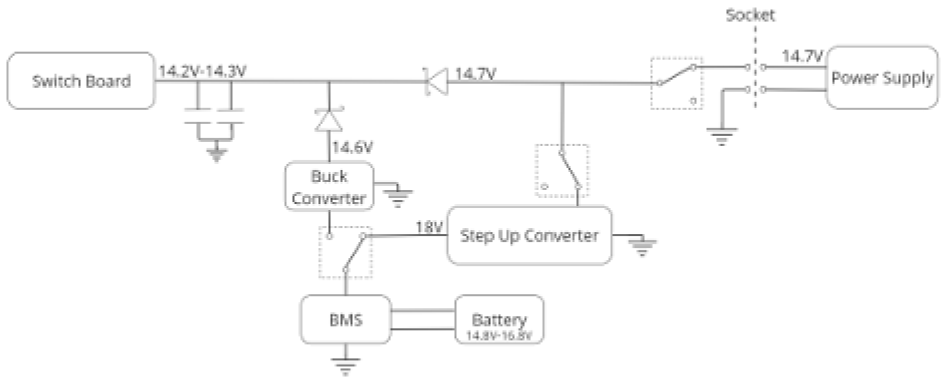
We designed the robot's power management system to monitor its battery levels during operation. A voltage sensor integrated into the system alerts the robot when the battery level drops below a predefined threshold. Upon detecting low battery levels, the robot navigates autonomously to a charging dock station, using the navigation stack. The robot aligns with the docking station through an IR emitter-based guidance system to establish a secure connection for charging.

The dock station operates on a wall power supply (230V AC) and outputs a maximum of 14V, 30A DC. When in charging mode, the robot is powered directly by the dock station’s supply while simultaneously charging its battery. A battery management system (BMS) ensures overcharge and discharge protection during this process. A relay-based switching system facilitates transitions between charging and discharging states.

The power management system designed as a dual-source system, ensuring the robot is always powered either by the battery or the dock station. The robot uses a 4-cell LiFePO4 battery with a capacity of 10,000mAh, providing up to 45 minutes of operation on a full charge. We selected LiFePO4 over lithium-ion and lead-acid batteries due to its superior discharge rate, lower bulk, and compatibility with the robot’s voltage and performance requirements.

To meet the charging conditions of battery management system (14V to 18V), we employed a 600W step-up converter to boost the dock station's output voltage during charging. Additionally, a 600W step-down converter regulates the battery's output voltage during discharging, ensuring a steady 14V supply to meet the robot's power requirements. This regulation is crucial since the battery’s voltage can fluctuate between 16.5V when fully charged and lower levels during discharge.

During charging, the BMS draws a maximum of 10A from the dock station while also powering the robot. Conversely, in discharging mode, the robot powered exclusively by the battery. The dual-source circuit guarantees continuous operation, regardless of the robot's state. A schematic diagram of the dual-source system is shown below.


Socket
Switch Board
14.2V-14.3V
14.7V
14.7V
Power Supply
14.6V
Buck Converter
18V
Step Up Converter
BMS
Battery
14.8V-16.8V