**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 [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 ensure the dataset was diverse and robust, we applied various data augmentation techniques such as rotation, contrast adjustment, brightness variation, noise injection, and flipping. These augmentations expanded the dataset’s coverage of potential light conditions and addressed challenges associated with the robot’s actual camera input. The final dataset included 4,796 images, distributed across three subsets: 4,003 images in the training set, 397 images in the test set, and 196 images in the validation set.

To maintain the integrity of performance evaluations, we ensured that the test and validation sets were not augmented and did not overlap with the training set. Using this well-structured dataset, we trained the YOLOv10-small model, achieving notable performance metrics that demonstrated the model's ability to detect and categorize human agents effectively. During training, 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.

The trained model detects human agents and their respective classes **Figure 1, Figure 2** in each video frame, producing bounding box coordinates and class labels for each agent. To integrate this information into the robot's dynamic environment map, we transform the 2D coordinates from the video frames into a 3D reference frame. Then, we cross-reference these transformed coordinates with human agents identified through LiDAR data, enabling unique identification of each agent and their assigned class within the environment.

A group of people walking in a mall

Description automatically generated

Figure 1

A person and person wearing face masks

Description automatically generated

Figure 2

#### 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 **figure 3**.

A diagram of a step up converter

Description automatically generated

**figure 3**

**Evaluation**

**Human Detection and Categorization Module**

We validated the YOLOv10-small model using the test dataset after completing its training on the training set. The validation demonstrated reasonable precision and recall metrics for each class, as presented in **Table 1.**

**Table 1**

| **Class** | **Images** | **Instances** | **Precision** | **Recall** | **Mean Average Precision (mAP) at IoU=0.5** | **Mean Average Precision (mAP) at IoU=0.5:0.95** |
| --- | --- | --- | --- | --- | --- | --- |
| All | 397 | 1296 | 0.71 | 0.74 | 0.76 | 0.52 |
| Child | 127 | 333 | 0.75 | 0.83 | 0.86 | 0.55 |
| Normal-adult | 207 | 631 | 0.66 | 0.62 | 0.66 | 0.65 |
| Elder | 102 | 163 | 0.65 | 0.64 | 0.67 | 0.52 |
| Disabled | 148 | 169 | 0.76 | 0.85 | 0.87 | 0.65 |

The metrics in **Table 1** show fair and reasonable precision, recall, and mean average precision (mAP) values for each class. During training, we analyzed the training losses and validation losses, including box regression loss, classification loss, and distribution focal loss (DFL), against the training epochs to identify the optimal set of weights. **Figure 4** illustrates these loss plots along with precision and recall values across epochs.

**Figure 4**

A group of graphs showing different types of data

Description automatically generated with medium confidence

We tested the trained model on video sources to evaluate its ability to detect and classify human agents**. Figures 1 and 2** demonstrate the model's performance in real-time environments. The results show that the model accurately detects human agents and classifies them into the predefined categories even in dynamic scenarios.

To further evaluate performance, we compared the YOLOv10-small model against other models trained on the same dataset. **Table 2** presents the observations, highlighting differences in average precision (AP) and latency across models.

Table 2

| **Model** | **Parameters (M)** | **Average Precision (AP)** | **Latency (ms)** |
| --- | --- | --- | --- |
| YOLOv10-Nano | 2.3 | 0.41 | 1.28 |
| YOLOv10-Small | 7.2 | 0.76 | 2.8 |
| YOLOv10-Mega | 15.4 | 0.78 | 4.9 |
| YOLOv8-Small | 11.3 | 0.64 | 7.4 |
| RTMDet | 8.3 | 0.42 | 6.6 |

Based on these results, we determined that the YOLOv10-Mega model, although achieving the highest precision, was unsuitable for real-time applications due to its high latency. Conversely, the YOLOv10-small model offered an optimal balance between precision and latency, making it the most practical choice for accurately detecting and categorizing human agents in dynamic environments.

Related work

In the human detection and classification there are several approaches are presented in the literature. Here the study [alphapose 1] uses a camera feed of the environment and identifying human agents and their features the model track their movement of hole body. But this feature extraction and tracking take high computational overhead making the high latency. So this approach is not suitable for the real time human detection and categorization based on their movement. Then the study [2 mivolo] proposed a method to detect human agents and their gender and age. But using this model it can only detect the age and gender. But when it comes to movement of human agents there will be more features affecting. Low configurability make this model is not suitable for human agent detection with knowledge about their movement details for human path planning. Also the study [3 dpnet] proposed a method to object detection with lightweight attention. Also study [4 yolo v10] proposed a lightweight method for object detection wich makes possible real time operation and configurable for detect custom classes detection.

### Related Work

Several approaches for human detection and classification have been explored in the literature. [alphapose 1] utilizes a camera feed to identify human agents and extract their features, enabling the tracking of whole body movements. However, the high computational overhead of feature extraction and tracking leads to significant latency, making it unsuitable for real-time detection and categorization based on movement. [2 mivolo] proposes a method focused on detecting human agents' gender and age, which is effective for these specific attributes but fails to account for other features influencing movement. Its limited configurability reduces its effectiveness for applications requiring detailed movement analysis, such as human path planning. In contrast, [3 dpnet] introduces an object detection approach leveraging lightweight attention mechanisms to improve efficiency, while [4 yolo v10] offers a lightweight object detection framework designed for real time operation. Its flexibility in detecting custom classes makes it a robust and adaptable solution for human detection and categorization tasks.

[1] Fang, Hao-Shu, Jiefeng Li, Hongyang Tang, Chao Xu, Haoyi Zhu, Yuliang Xiu, Yong-Lu Li, and Cewu Lu. "Alphapose: Whole-body regional multi-person pose estimation and tracking in real-time." IEEE Transactions on Pattern Analysis and Machine Intelligence 45, no. 6 (2022): 7157-7173.

[2] Kuprashevich, Maksim, and Irina Tolstykh. "Mivolo: Multi-input transformer for age and gender estimation." In International Conference on Analysis of Images, Social Networks and Texts, pp. 212-226. Cham: Springer Nature Switzerland, 2023

[3] Zhou, Quan, Huimin Shi, Weikang Xiang, Bin Kang, and Longin Jan Latecki. "DPNet: Dual-path network for real-time object detection with lightweight attention." IEEE Transactions on Neural Networks and Learning Systems (2024).

[4] Wang, Ao, Hui Chen, Lihao Liu, Kai Chen, Zijia Lin, Jungong Han, and Guiguang Ding. "Yolov10: Real-time end-to-end object detection." *arXiv preprint arXiv:2405.14458* (2024).

Conclusion

In this study we developed a receptionist robot that can do the crowd navigation with understanding about the human agent factors that affect to their movements in dynamic environment, able to use elevator and perform the multi floor navigation and can do the self charging itself.

In the crowd navigation we enhance the human agent path planning using the human preferred velocities and human classes that affect to the movement of human agents. Based on these factors the algorithm generate a footprint that reflect the uncertain factors that enhance the social and safe path planning for robot that lead to social and safe navigation trough crowd.

In multi-floor navigation, we extend 2D navigation algorithms to enable 3D navigation improving real time performance in limited computational capabilities. Also, we enhanced the recovery options for time critical interactions like elevator loading for improved performance. In robot-elevator interaction we have integrate the elevator button detection, localization capability to the robot. And also we have implemented a four degrees of freedom robotic arm to have a more precise movement when precessing the elevator button, making sure the effective robot-elevator interaction.

This study presents the development of a receptionist robot that can navigate through crowds, use elevators to perform multi-floor navigation, and autonomously self-charge. In the implementation of crowd navigation, we utilized human preferred velocities and movement characteristics based on human classes to account for factors influencing human motion in dynamic environments. This approach enabled the algorithm to generate uncertainty-aware footprints, ensuring socially acceptable and safe path planning for the robot. For multi-floor navigation, we extended traditional 2D navigation algorithms to support 3D navigation, achieving real-time performance even with limited computational resources. Enhanced recovery mechanisms to optimize the handling time-sensitive tasks such as elevator loading, improving the robot’s efficiency in critical scenarios. Additionally, implemented elevator button detection and localization capabilities, supported by a four-degrees-of-freedom robotic arm for precise interaction with elevator buttons. These integrated advancements allow the robot to operate across diverse and complex environments.

[1] Wang, Ao, Hui Chen, Lihao Liu, Kai Chen, Zijia Lin, Jungong Han, and Guiguang Ding. "Yolov10: Real-time end-to-end object detection." *arXiv preprint arXiv:2405.14458* (2024).

[2] Sharma, Devansh, Tihitina Hade, and Qing Tian. "Comparison of deep object detectors on a new vulnerable pedestrian dataset." *arXiv preprint arXiv:2212.06218* (2022).