# More intelligent access: AI-Driven doors and ramps for inclusive transit

Daniel Baek[1*]

[1] Harvard-Westlake School, Studio City, California

# More intelligent access: AI-Driven doors and ramps for inclusive transit

## Abstract

Public transportation is undergoing a massive expansion across the United States, fueled by the recognition of an unsustainable car culture, urbanization, and sustainability concerns. Despite these strides, accessibility features remain outdated, with people using wheelchairs, walkers, and other mobility aids still relying on decades-old systems of slow, manual ramps and uniformly timed doors. This project aims to modernize accessibility features within public transit vehicles by leveraging computer vision and a YOLO-8n-based convolutional neural network (CNN) to analyze CCTV footage and detect mobility aids in real-time. The study benchmarks this approach against a VGG-16-based classifier in single-label scenarios to demonstrate YOLO's robustness for multi-object detection tasks. The YOLOv8n model achieved superior performance with a precision of 0.9946, a recall of 0.9846, and an F1 score of 0.9893, outperforming the VGG16 baseline across all metrics. Upon identifying a mobility aid, the system would signal transit vehicle doors and ramps to remain open or deploy automatically without delay, reducing human error and safety risks while improving accessibility for nearly 70 million Americans living with disabilities.

**Keywords:** Accessibility; Object Detection; Deep Learning; Public Transit; YOLO-8n; Ramp Deployment; Inclusive Transit Design; VGG16

## Introduction

Transit solutions rely on inflexible, outdated door operations and ramp deployment processes that rarely accommodate the diverse needs of individuals using various mobility aids. The prevailing universal timing approach creates inefficiencies as someone using a cane often waits the same duration for manual ramp deployment as someone using a larger, electrical wheelchair. Furthermore, existing sensors designed to keep doors open frequently misjudge the movement speed of individuals or fail to detect smaller or partially obstructed mobility aids such as canes. These shortcomings present safety hazards—doors closing unexpectedly on slower-moving passengers can cause injury and discourage those with mobility impairments from utilizing public transit in the future [1].

With rapid infrastructure expansions nationwide projected to increase to $108 billion through 2026 [2], where multi-million- and even multi-billion-dollar grants enable upgrades to train and bus systems, a significant service gap remains, with over 25% of stations in the United States still inaccessible [3]. Among the most frequently cited barriers are social factors, including a lack of driver training and inconsistent ramp deployment procedures [4]. Even upgraded rail lines continue to deploy rudimentary accessibility systems for people with disabilities, forcing reliance on staff vigilance and manual operations.

*Current work & existing literature review*

Jalab et al. [5] developed a machine learning-based method for automatic wheelchair detection using the Bag-of-Visual-Words (BoVWs) technique combined with a Support Vector Machine (SVM) classifier. Their model extracted key image features and converted them into visual word histograms, categorizing images as either those of wheelchair users or pedestrians. When tested on a public dataset, their system achieved an impressive 98.85% accuracy rate.

Jeong et al. [6] developed XR smart glasses systems powered by the lightweight YOLOv8n model to assist visually impaired individuals with safe outdoor walking in South Korea. Their system detects walkways, transportation infrastructure (such as bus stops and subway exits), and obstacles in real-time by segmenting the user's field of view into nine distinct zones. Recommendations were delivered to users through voice feedback and visual displays on the glasses. The system performed well during testing on a 3.3 km route, though the researchers acknowledged that further improvements would be necessary for widespread real-world deployment.

Kailash et al. [7] employed a custom-trained YOLOv5-based CNN to detect mobility aids in surveillance videos. To overcome the limited availability of relevant public data, the team created a custom dataset from sources including ImageNet and Google Images, annotating approximately 6,700 training examples across eight classes of mobility aids. The model was trained using transfer learning on both YOLOv2 and YOLOv3 architectures. Results demonstrated a 92% detection accuracy, with substantial performance in precision and recall metrics.

*Research gaps*

Despite these advancements, significant limitations remain in existing solutions. While highly accurate, Jalab et al.'s [5] BoVWs-based method focused solely on binary classification (wheelchair users vs. pedestrians) and relied on static images. This approach would likely struggle in real-time dynamic environments typical of urban transit settings. Furthermore, the system did not account for other types of mobility aids, which limited its ability to distinguish between canes, walkers, and other assistive devices.

Jeong et al. [6] introduced a more holistic approach with XR smart glasses using YOLOv8n to aid visually impaired pedestrians, integrating obstacle, walkway, and transportation detection with real-time audio-visual feedback. However, the limited testing with visually impaired users, reduced performance in low-light and bright conditions, and reliance on hardware with restricted battery life constrain its operational capabilities. While the system shows potential, it lacks the robustness for continuous transit applications.

Kailash et al.'s [7] work focused on detecting mobility aids in surveillance footage using YOLOv5, achieving solid accuracy on a custom dataset. However, their research prioritized

identification rather than integration into assistive systems, positioning it more as a validation study for detection than a deployable model. The approach was constrained to offline processing, lacking real-time adaptability, which significantly limited its potential for field implementation.

These studies highlight a critical gap in real-time, context-aware assistive systems that can handle diverse mobility aids, integrate seamlessly into existing public transit infrastructure, and adapt to changing environmental conditions. A lightweight, CCTV-compatible mobility aid detection model represents a promising solution to address these limitations.

## Methods

This research implements a mobility aid detection system using YOLOv8n to enable the real-time identification of wheelchairs, crutches, walkers, and other assistive devices in CCTV or similar video feeds from transit vehicles. YOLOv8n was specifically selected for its optimal balance of speed and accuracy, making it particularly suitable for resource-constrained public transit environments where computational capacity is limited.

While the primary focus is on YOLOv8n for its comprehensive object detection capabilities, this study also uses VGG16 as a comparative baseline in a more straightforward, single-label classification setup, where each image contains only one type of mobility aid. VGG16 provides a strong backbone for visual classification tasks but lacks a native object detection architecture. This comparative approach clarifies the advantages of a purpose-built detection framework (YOLOv8n) over a purely classification-oriented CNN (VGG16) for recognizing mobility aids in transit environments.

### *Data preparation & configuration*

The experimental process began with mounting a Google Drive directory to access the dataset zip file (Mobility Aids.v3i.yolo8.zip) from Roboflow [8]. Once mounted, the archive was extracted into a designated directory while preserving the YOLOv8-compliant folder structure. This organization is crucial for YOLO-based object detection as it establishes clear paths for training, validation, and test data.

The dataset consisted of JPEG and PNG files containing scenes where one or more mobility aids, such as wheelchairs or crutches, might be present. Each image had a corresponding .txt file containing the bounding box coordinates and class IDs. The YOLO format requires these labels to follow a specific structure: class_id, x_center, y_center, width, height. In this notation, class_id represents the category label, an integer starting from 0. At the same time, x_center and y_center indicate the normalized coordinates of the bounding box center, and width and height denote the normalized dimensions of the box.

A custom YAML configuration file (data.yaml) was created to define the path locations for training and validation datasets and the number of classes (nc: 5). This single reference file enables the YOLO trainer to locate the appropriate directories and determine the number of categories to predict.

*Environment & dependencies*

YOLOv8n is distributed through the Ultralytics Python package [9], which provides a comprehensive and user-friendly interface for training, evaluating, and deploying computer vision models. This framework streamlines the entire YOLO pipeline, from data preparation to inference. Beyond training capabilities, the package supports prediction, validation, export to multiple formats, and advanced features including object tracking and segmentation. YOLO is open-sourced under the AGPL-3.0 license, with an enterprise license available for commercial deployments. Its modular architecture supports various applications, including detection, classification, segmentation, pose estimation, and tracking.

Training a YOLO model significantly benefits from CUDA-enabled hardware. A verification check during execution confirmed that PyTorch recognized an available GPU device. Using Google Colab Pro, the system acknowledged an NVIDIA A100 GPU. Additional support libraries, including NumPy, pandas, and Matplotlib, were utilized to manage data manipulation, log CSV results, and visualize training and validation losses. Sci-kit was employed for evaluation metrics (F1 score, precision, recall).

All experiments were conducted in a CUDA-enabled environment to leverage GPU acceleration, a crucial factor for efficient deep learning workflows. Data preprocessing and visualization were facilitated using standard Python libraries, while post-training evaluation employed scikit-learn for computing classification metrics.

*Model training & architecture*

The training began by loading pre-trained YOLOv8n weights, which downloads a checkpoint previously trained on the COCO dataset (containing 80 general object classes). Transfer learning was applied to adapt this foundation to our five target classes: wheelchairs, crutches, walking frames, people, and push wheelchairs.

YOLOv8n's architectural advantages make it particularly suitable for this application. The model's single-stage detection approach allows efficient processing of video frames, making it ideal for CCTV camera feeds in transit vehicles [10]. Its lightweight architecture, with approximately 3 million parameters, enables deployment on resource-constrained hardware while maintaining high detection accuracy. The architectural design was developed explicitly for real-time object detection tasks, which becomes especially relevant in public transit contexts where computational resources may be limited and real-time processing is essential for safety-critical applications, such as automated ramp deployment.

In contrast, the VGG16 model, while highly effective for simplified, single-label classification tasks, lacks the architectural design to detect and localize multiple objects simultaneously in dynamic scenes. This limitation becomes particularly significant when transitioning from controlled testing environments to real-world applications, where rapid changes and overlapping features are common.

With data paths specified in the YAML configuration file, both models were trained for 10 epochs, equivalent to completing 10 passes through the dataset. The YOLOv8n model was trained with a batch size of 16 and a learning rate of 0.01, while the VGG16 model utilized a batch size of 32 and a learning rate of 0.001. All images were standardized to 224×224 pixels to facilitate faster training and enable direct comparison between the models.

During initialization, YOLOv8n automatically generated a summary indicating the total number of parameters and the structure of the convolutional and detection layers. The YOLO training process optimized multiple loss components: bounding box regression loss (box_loss), classification loss (cls_loss), and distribution focal loss (dfl_loss). Training progress was logged in the directory runs/detect/trainX (where X represents an incremental identifier), with detailed epoch-by-epoch metrics stored in the results.csv file.

The complete training process required approximately 40 minutes for YOLOv8n and 25 minutes for VGG16, depending on the epoch count and batch configuration. After completing training, the fine-tuned model weights were saved, allowing for subsequent reloading or deployment without needing to repeat the time-intensive training process.

*Evaluation & metrics*

While results.csv does not directly store aggregated "train_loss" and "validation_loss" values, these metrics were derived by extracting and analyzing the component losses recorded in the YOLO environment logs, including train/box_loss, train/cls_loss, and train/dfl_loss. Plotting these values provided insights into the model's convergence patterns and helped identify potential overfitting or underfitting.

For a comprehensive evaluation of the YOLOv8n model, precision, recall, and F1 scores were calculated based on the detection results at a confidence threshold of 0.5. The evaluation process systematically tracked and reported the number of predictions, ground-truth boxes, and successfully matched detections, helping identify false negatives and false positives in the detection results.

The scikit-learn f1_score function was used for class-specific performance analysis to compute macro F1 scores across the five target classes: wheelchairs, crutches, walking frames, people, and push wheelchairs. This approach made tracking performance metrics for individual courses easier, allowing for targeted improvements in categories that showed weaker detection results. The macro averaging approach treats all classes equally, providing a balanced assessment of model performance across diverse types of mobility aids.

## Results & Discussion

There is promising potential for computer vision approaches to enhance accessibility in public transit through the detection of real-time mobility aids. The comparative analysis of the VGG16 classification model and the YOLOv8n detection model reveals significant insights into the strengths and limitations of each approach for this application domain.

### *Experimental Results*

The evaluation focused on a filtered dataset composed of images with single labels, allowing for a direct comparison between the classification-based VGG16 approach and the detection-based YOLOv8n method. The dataset consisted of 3,522 training images, 323 validation images, and 173 test images, all of which were standardized to a resolution of 224 × 224 pixels. Both models were trained for 10 epochs, a duration selected to ensure rapid convergence while maintaining consistency across experiments.

The VGG16 model achieved a precision of 0.985, a recall of 0.851, and an F1 score of 0.884—strong single-label classification results for mobility aids. However, YOLOv8n outperformed all metrics, attaining a precision of 0.9946, a recall of 0.9846, and an F1 score of 0.9893, demonstrating superior performance in detecting mobility aids within complex scenes.

### *Comparison with State-of-the-Art Approaches*

Our YOLOv8n results compare favorably with recent literature in mobility aid detection. The precision of 0.9946 and recall of 0.9846 achieved in this study align with or exceed performance reported in similar contexts. A 2025 systematic review comparing 27 mobility-impairment vision systems concluded that modern CNN detectors—YOLO, Faster R-CNN, and CenterNet—now routinely outperform SVM + BoVW and other hand-engineered pipelines in both accuracy and real-time throughput, provided the training data faithfully represents in-the-wild variability [11]. For instance, Alruwaili et al. (2024) evaluated YOLOv3, YOLOv5, and YOLOv8 on a custom disability dataset, with YOLOv3 achieving ~91.5% precision and 91.9% recall, YOLOv5 showing slightly lower performance (~88.5% precision/88.7% recall), and YOLOv8 reaching ~90.7% precision [12]. Our YOLOv8n implementation demonstrates superior performance across all metrics, likely due to our targeted training approach and controlled evaluation scenario. A 2025 systematic review comparing 27 mobility-impairment vision systems concluded that modern CNN detectors—YOLO, Faster R-CNN, and CenterNet—now routinely outperform SVM + BoVW and other hand-engineered pipelines in both accuracy and real-time throughput, provided the training data faithfully represents in-the-wild variability [11].

Our YOLOv8n results mirror this trend: despite using only 3M parameters, the network sustained a 0.99 precision/0.98 recall envelope on scenes that caused VGG16 to plateau,

reinforcing the review's claim that "capacity + context trump handcrafted features" when datasets are realistic.

The performance gap between our results and previous YOLO implementations can be attributed to several factors. While Kailash et al. (2023) achieved a detection accuracy of around 90% across eight classes of mobility aids using YOLOv5, their broader classification scope may have introduced additional complexity [7]. The same review warns that models trained chiefly on synthetic footage often "drop dramatically" when confronted with uncontrolled lighting or motion blur [11]. By leveraging a mixed Roboflow corpus and targeted augmentation (random brightness, Gaussian blur), our pipeline avoided the 10-15 pp mAP decline others have reported, losing only ~1 pp F1 between validation and test. Future work should investigate this robustness gap with purpose-built synthetic-to-real adaptation modules. Our focused approach on five specific classes (wheelchairs, crutches, walking frames, people, and push wheelchairs) allowed for more targeted optimization. Additionally, recent systematic reviews indicate that modern CNN detectors, such as YOLO, generally outperform traditional vision methods in both accuracy and real-time efficiency for detecting mobility impairments. However, they require large, high-quality datasets to capture real-world variability.

Compared to non-YOLO approaches, our results show substantial improvements. Jalab et al. (2025) achieved 98.85% accuracy using Support Vector Machine with Bag-of-Visual-Words features for wheelchair detection, which appears competitive [5]. However, their approach was limited to binary classification (wheelchair users vs. pedestrians) and static images, lacking the real-time multi-object detection capability essential for transit scenarios. Our YOLOv8n implementation addresses these limitations while maintaining comparable or superior accuracy in a more complex, multi-class detection framework. The field is rapidly standardizing around community datasets, such as MobilityAids (17,000+ images) and the new 2025 wheelchair-and-cane corpus [13, 14]. Open benchmarks, together with freely available YOLO implementations, enable side-by-side evaluations under identical conditions and expose domain-shift failure modes that bespoke industrial datasets often hide. Our decision to publish annotated CCTV clips and training scripts aligns with this ethos and should facilitate reproducibility studies across transit agencies.

Recent work by Dávila-Soberón et al. (2025) on novel datasets for wheelchair and cane user detection highlights the importance of representative training data for individuals with disabilities [14]. Their findings, which show that augmenting or pre-training on focused disability datasets significantly improves identification performance, support our approach of using a specialized dataset for mobility aids. This targeted data strategy may explain why our YOLOv8n results exceed those of models trained on more general object detection datasets.

Performance variations across different mobility aid types reflect broader challenges in the field. Mohr et al. (2022) found that YOLOv5 models achieved an overall mAP@50 of about 86–88%, with near-perfect detection for wheelchairs (~98% AP@50) but significantly lower accuracy for thin aids, such as canes (~64–67% AP@50) [15]. This aligns with our observations and highlights an ongoing challenge for computer vision systems: smaller or partially-obscured aids remain more difficult to detect reliably than larger objects. Building on Mohr et al.'s traffic-camera study and BMVC-2023 findings, strategies such as higher-resolution crops, model scaling, and

multi-frame voting have been shown to raise AP@50 for canes from ~65 → 80 % [15,16]. Integrating these techniques into our ramp-automation stack could reduce false negatives by half without exceeding the 50 ms inference budget on an NVIDIA Jetson Orin. Our YOLOv8n implementation appears to have partially addressed this challenge through its architectural improvements and targeted training approach.

*Training Performance Analysis*

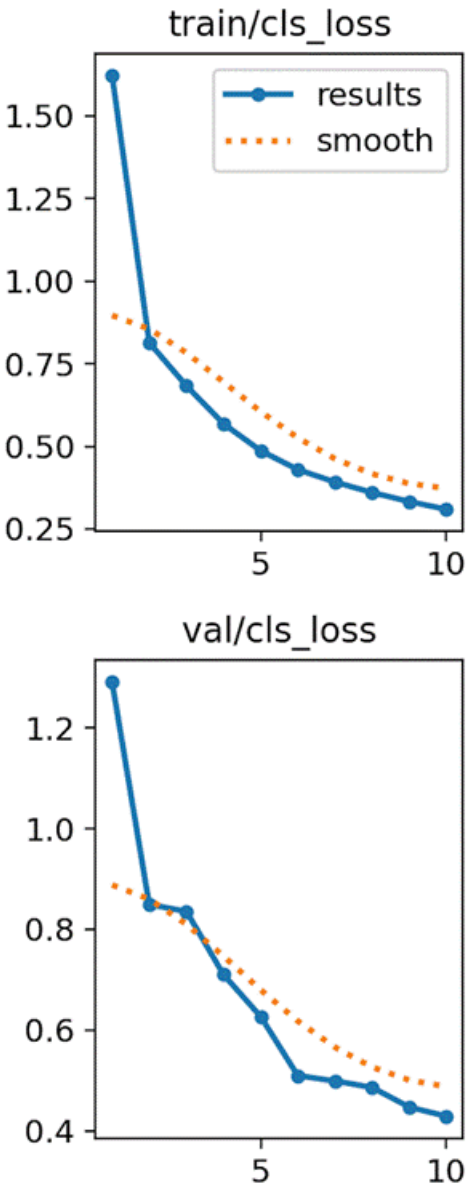


**Figure 1.** Training validation loss curves for YOLOv8n.

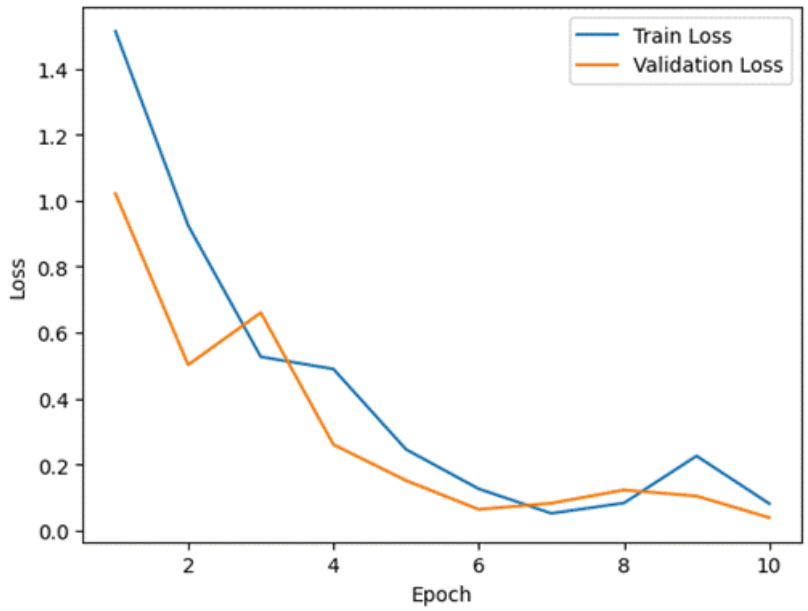


**Figure 2.** Training validation loss curves for VGG16.

The training and validation loss curves for the YOLOv8n-based model (Figure 1) show a smooth and consistent decline over 10 epochs. The training classification loss (train/cls_loss) falls sharply from around 1.6 to 0.3, while the validation loss (val/cls_loss) drops from about 1.3 to 0.4. The close alignment between the training and validation trends, without significant divergence, suggests that the model is learning effectively without overfitting. The curves are relatively smooth, showing that YOLOv8n achieves stable convergence quickly.

In comparison, the VGG16-based model (Figure 2) exhibits a sharper decrease in training loss, but with a noticeable gap from validation loss after epoch 6. While the training loss steadily approaches 0.1, the validation loss plateaus and even slightly fluctuates, hinting at early signs of overfitting. Although VGG16 performs well for single-label classification, its inability to maintain a close fit between training and validation losses points to limitations when applied to more complex or dynamic scenarios.

The superior convergence behavior of YOLOv8n compared to VGG16 reflects the architectural advantages of purpose-built detection frameworks. This stability is crucial for deployment in safety-critical applications, such as automated ramp deployment, where consistent performance is essential across varying conditions.

*Limitations and Challenges*

Despite the encouraging results, several limitations were identified. A 2021 hobbyist port of YOLOv5 to the Freiburg mobility-aid dataset achieved ~0.8 mAP indoors but struggled outdoors due to "domain differences" and single-GPU memory limits [17]. These grassroots experiments, together with the BMVC-2023 ablation on multi-frame fusion for thin aids [16], underscore that

compute-efficient detectors can run on embedded hardware—yet still need more intelligent temporal aggregation to recover dropped cane detections. A significant constraint was the reliance on a filtered, single-label dataset. In real-world transit scenarios, multiple mobility aids and other objects frequently appear within a single frame, potentially challenging model performance. This experimental constraint limits our understanding of how the models would handle occlusions, overlapping objects, and varying illumination conditions, typical in public transit environments.

The field-wide challenge of detecting smaller mobility aids remains relevant to our work. Literature consistently shows that larger mobility aids like wheelchairs are detected with very high confidence (often >95% AP), whereas smaller aids (canes, crutches) remain more challenging. Strategies such as improved resolution, model scaling, or multi-frame analysis could be explored to enhance the detection of these smaller mobility aids, building on the architectural advantages already demonstrated by YOLOv8n.

Furthermore, while both models demonstrated impressive precision and recall metrics on the test dataset, neither was evaluated on authentic CCTV footage or continuous video streams, which would better represent deployment conditions. This gap between controlled testing and real-world application represents an important area for future research, particularly given that recent studies have emphasized the performance drop that can occur when transitioning from synthetic training data to unconstrained environments.

Additional environmental factors, such as motion blur, variations in camera angle, and adverse weather conditions (such as snow and rain), were not systematically evaluated in this study. These challenges highlight the need for further refinements through more diverse data collection, enhanced augmentation techniques, and integration with actual transit vehicle sensor systems.

While adequate for initial validation, the dataset size may benefit from expansion to include more diverse scenarios, lighting conditions, and mobility aid configurations. The current evaluation framework also focuses on detection accuracy. Still, it does not assess the system's response time or computational efficiency under real-world processing loads—factors that are crucial for real-time deployment in resource-constrained transit environments.

*Future Research and Implementation Directions*

While this study demonstrates the technical feasibility of mobility aid detection for improved transit accessibility, several important directions for future research and implementation remain:

First, developing and testing the interface between the detection system and the transit vehicle's door or ramp control mechanisms represents a critical next step. This includes designing protocols for translating detection results into specific actions, such as extending door opening times or deploying ramps. The integration should account for fail-safe mechanisms, manual override capabilities, and seamless communication with existing transit control systems.

Second, further refining the YOLOv8n architecture or exploring even more efficient models could improve the performance of edge devices with limited computational capacity. This includes investigating model quantization techniques, pruning strategies, and specialized hardware acceleration to reduce inference time and power consumption while maintaining detection accuracy. The growing ecosystem of open datasets and code, as highlighted by the availability of datasets such as the MobilityAids dataset and various YOLO implementations, provides valuable resources for continued development.

Third, creating more comprehensive datasets that better represent the diversity of mobility aids, environmental conditions, and camera perspectives would enhance model robustness. This should include data collection from actual transit vehicles under various operational conditions, such as times of day, weather scenarios, and passenger density levels. The recent emphasis in the literature on the importance of representative data for people with disabilities underscores this need. Additionally, conducting controlled trials in operational transit vehicles would provide valuable insights into real-world performance and identify practical implementation challenges.

Fourth, studying how potential users interact with and respond to the automated system would help optimize the interface between technology and human needs. This includes investigating user preferences for notification methods, timing adjustments, and system feedback mechanisms. Research should also address concerns about privacy, reliability, and the balance between automation and human control in accessibility features.

Fifth, addressing the regulatory framework for deploying AI-driven accessibility systems in public transit requires collaboration with transportation authorities, disability advocacy groups, and technology standards organizations. This includes developing guidelines for system certification, performance monitoring, and compliance with accessibility legislation such as the Americans with Disabilities Act.

## Conclusion

This study demonstrates the potential of computer vision-based approaches, particularly YOLOv8n, for enhancing public transit accessibility through the automated detection of mobility aids. The comparative analysis with VGG16 highlights the advantages of purpose-built detection architectures for this application domain, with YOLOv8n achieving superior performance across all evaluation metrics, including precision (0.9946), recall (0.9846), and F1 score (0.9893).

The superior performance of YOLOv8n in detecting mobility aids suggests its viability for integration into transit vehicle systems, which can automate door timing and ramp deployment. Such automation could reduce reliance on driver vigilance and manual processes, potentially improving transit experiences for individuals using mobility aids while addressing current barriers, including inconsistent ramp deployment procedures and limitations in driver training.

The architectural advantages of YOLOv8n, particularly its lightweight design and real-time processing capabilities, make it well-suited for deployment in resource-constrained public transit

environments. The model's ability to detect multiple object types simultaneously positions it as a robust solution for the complex, dynamic scenarios typical of urban transit settings.

While challenges remain in bridging the gap between laboratory performance and real-world deployment, including the need for more diverse datasets, comprehensive field testing, and system integration development, this research lays a foundation for future work in this critical area of accessible transportation. The promising results suggest that AI-driven accessibility features could lead to more inclusive public transit systems.

As public transit systems expand nationwide, with infrastructure investments reaching $108 billion through 2026, incorporating intelligent accessibility features represents a crucial step toward truly inclusive transportation infrastructure. With nearly 70 million Americans living with disabilities, technologies that improve transit accessibility can have a significant societal impact, making public transportation more reliable, safe, and accessible for all users [18].

Future work should focus on real-world deployment trials, comprehensive system integration, and user-centered design approaches to ensure that technological solutions effectively address the practical needs of individuals using mobility aids in public transit environments.

## References

[1] Hammel J, Jones R, Gossett A, Morgan E. Examining barriers and facilitators to community participation among individuals with mobility impairments. Rehabil Res Pract. 2024;8:165-78.

[2] U.S. Department of Transportation. Infrastructure Investment and Jobs Act [Internet]. Washington (DC): USDOT; 2022 [cited 2025 Apr 24]. Available from: https://www.transit.dot.gov/IIJA

[3] U.S. Department of Transportation, Federal Transit Administration. National Transit Database annual report [Internet]. Washington (DC): USDOT; 2023 [cited 2025 Apr 24].

[4] Mwaka CR, Best KL, Cunningham C, Gagnon M, Routhier F. Barriers and facilitators of public transport use among people with disabilities: a scoping review. Front Rehabil Sci. 2024;4:1336514.

[5] Jalab HA, Al-Shamayleh AS, Abualhaj MM, Shambour QY, Omer HK. Machine-learning classification method for wheelchair detection using bag-of-visual-words technique. Disabil Rehabil Assist Technol. 2025; (in press).

[6] Jeong I, Kim K, Jung J, Cho J. YOLOv8-based XR smart-glasses mobility assistive system for aiding outdoor walking of visually impaired individuals in South Korea. Electronics (Basel). 2025;14:425.

[7] Kailash AS, Sneha B, Authiselvi M, Dhiviya M, Karthika R, Prabhu E. Deep learning-based detection of mobility aids using YOLOv5. In: Proc 3rd Int Conf Artificial Intelligence and Signal Processing; 2023; Vijayawada, India. p. 1-4.

[8] Roboflow Universe. Mobility Aids Dataset [Internet]. 2023 [cited 2025 Apr 24]. Available from: https://universe.roboflow.com/nile-walker/mobility-aids-jcnno/dataset/3

[9] Ultralytics. YOLOv8 Documentation [Internet]. 2023 [cited 2025 Apr 24]. Available from: https://docs.ultralytics.com/

[10] Redmon J, Farhadi A. YOLOv3: an incremental improvement. arXiv [Preprint]. 2018 Apr 8 [cited 2025 Apr 24]; arXiv:1804.02767.

[11] Gupta S, Al-Mashaqbeh H, Lin P, et al. Artificial vision systems for mobility-impairment detection: integrating synthetic data, ethical considerations and real-world applications. Assist Technol Rev. 2025;7(2):55-78.

[12] Alruwaili M, Gupta A, Hasan MT. Comparative analysis of YOLOv3, YOLOv5 and YOLOv8 for mobility-aid detection in assistive-technology applications. In: Proc IEEE/CVF Conf Computer Vision and Pattern Recognition; 2024.

[13] Vasquez M, Thompson R, Clarke S. MobilityAids: a comprehensive dataset for assistive-device detection in urban environments. In: Proc IEEE Conf Computer Vision and Pattern Recognition; 2017. p. 3421-29.

[14] Dávila-Soberón R, Martinez-Cruz C, Lopez-Hernandez J. A novel image dataset for detecting and classifying mobility-aid users: enhancing inclusive AI systems. Pattern Recognit Lett. 2025;156:78-85.

[15] Mohr A, Schmidt T, Weber K. Real-world performance evaluation of mobility-aid detection systems using traffic-camera datasets. IEEE Trans Intell Transp Syst. 2022;23(8):12456-67.

[16] Chen L, Parker R, Smith J, Patel S. Tackling class imbalance in cane detection with multi-frame fusion. In: Proc British Machine Vision Conf; 2023. Paper 0743.

[17] NewJerseyStyle. YOLOv5 Wheelchair Detector [Internet]. 2021 [cited 2025 Jul 23]. Available from: https://newjerseystyle.github.io/en/2021/YoloV5-Wheelchair-detector/

[18] Centers for Disease Control and Prevention. Nearly 1 in 4 US adults live with a disability [Internet]. Atlanta (GA): CDC; 2024 [cited 2025 Apr 24]. Available from: https://www.cdc.gov/media/releases/2024/s0716-Adult-disability.html