No Pedestrian Left Behind (NPLB): Real-Time Detection and Tracking of Vulnerable Road Users for Adaptive Traffic Signal Control

Anas Gamal Aly
Stetson University
DeLand, Florida, USA
agamal@stetson.edu

Hala ElAarag
Stetson University
DeLand, Florida, USA
helaarag@stetson.edu

Abstract

Current pedestrian crossing signals operate on fixed timing without adjustment to pedestrian behavior, which can leave vulnerable road users (VRUs) such as the elderly, disabled, or distracted pedestrians stranded when the light changes. We introduce **No Pedestrian Left Behind (NPLB)**, a real-time adaptive traffic signal system that monitors VRUs in crosswalks and automatically extends signal timing when needed. We evaluated five state-of-the-art object detection models on the BGVP dataset, with YOLOv12 achieving the highest mean Average Precision at 50% (mAP@0.5) of 0.756. NPLB integrates our fine-tuned YOLOv12 with ByteTrack multi-object tracking and an adaptive controller that extends pedestrian phases when remaining time falls below a critical threshold. Through 10,000 Monte Carlo simulations, we demonstrate that NPLB improves VRU safety by 71.4%, reducing stranding rates from 9.10% to 2.60%, while requiring signal extensions in only 12.1% of crossing cycles.

CCS Concepts

• **Computing methodologies** → **Object detection**; **Tracking**; • **Computer systems organization** → **Embedded systems**; • **Applied computing** → *Transportation*; • **Mathematics of computing** → *Markov-chain Monte Carlo methods.*

Keywords

pedestrian detection, vulnerable road users, adaptive traffic signal control, computer vision, object tracking, AI for social good, computer vision, real-time systems

ACM Reference Format:
Anas Gamal Aly and Hala ElAarag. 2026. No Pedestrian Left Behind (NPLB): Real-Time Detection and Tracking of Vulnerable Road Users for Adaptive Traffic Signal Control. In *2026 ACM Southeast Conference (ACMSE 2026), April 23–25, 2026, Troy, AL, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/YYYYYYY.YYYYYYY

1 Introduction

Pedestrian safety remains a critical challenge for modern cities. The World Health Organization estimates that more than 270,000 pedestrians die annually in road traffic crashes, representing nearly 22% of global road fatalities [29]. Urban intersections are particularly risky because fixed-time pedestrian phases assume uniform walking speeds; in reality, speeds vary substantially across individuals and are often lower for elderly, disabled, or visually impaired pedestrians. This creates a risk that some pedestrians are still in the crosswalk when the signal expires. Evaluations of mid-block pedestrian signals (MPS) have shown measurable reductions in pedestrian–vehicle conflicts, reinforcing that timing aligned to observed behavior can improve safety [4].

Abdelrahman et al. [2] introduced VTPM, an LLM framework that generates real-time reports with 0.33 second latency. There are various machine learning methods that could be applied to Intelligent Transportation Systems (ITS) [12]. For traffic-flow detection, methods such as k-nearest Neighbor (k-NN), support vector regression (SVR), SVM, or long short-term memory (LSTM) were used. For accident detection and prevention, methods such as regression tree (CART, k-NN) [12] were used. Mohandas et al. [18] defines smart cities as the establishment of all the facilities required in the context of demand-based needs.

Machine learning methods such as Neural Networks have been found to be useful in improving the energy efficiency of street lights and creating weather-aware methods [9, 15, 26]. To implement smart city lights, Kama et al. [14] introduced solar street lights that collect data in real time on an online server to help plan the cleaning of the solar panels.

ACMSE 2026, April 23–25, 2026, Troy, AL, USA

ACM ISBN 979-8-4007-2062-8/2026/04
https://doi.org/10.1145/YYYYYYY.YYYYYYY

2 Related Works

Adaptive Signal Control. Traditional practice in pedestrian signal design relies on assumed walking speeds, which often under-serve slower pedestrians. Muley et al. [20] reviewed global practices and highlighted the limitations of fixed-time approaches. Adaptive signal control strategies have been shown to improve multimodal performance; for example, Akyol et al. [5] demonstrated that adjusting green times dynamically based on demand improved both throughput and pedestrian safety. Field evaluations of MPS report reduced probabilities of moderate and serious conflicts in before-and-after studies [4]. Networked sensor-based adaptive controllers also show potential for scaling adaptive strategies across corridors and cities [27]. Deploying SCOOT algorithm for adaptive signal timings improved the pedestrian travel times by 30% while increasing vehicle delays by 11% [5]

Prediction Models for Pedestrian Intention. Several recent studies move beyond detection to forecasting pedestrian behavior. Abdelrahman et al. [3] introduced VRUCrossSafe, a framework that predicts the crossing intention of pedestrians, cyclists, and scooter riders in real time, achieving high accuracy and inference speed. Such predictive models highlight the potential of CV-based monitoring but stop short of integrating predictions into traffic control. Our work builds on this literature by coupling real-time perception with a rule-based controller that dynamically extends pedestrian phases, addressing an important research gap.

YOLO (You Only Look Once) models offer fast, accurate pedestrian detection, while trackers such as DeepSORT [28], OC-SORT [10] and ByteTrack [30] maintain consistent pedestrian IDs across frames. Together, these tools allow estimation of walking speed and crossing progress directly from video feeds. Beyond detection, recent works have applied CV to predict pedestrian crossing intention, such as VRUCrossSafe, which achieved 94.7% accuracy at real-time speed by leveraging pose and appearance features [3].

Despite this progress, much of the literature optimizes signals primarily for vehicles or focuses on predicting pedestrian behavior without adapting the signal timing itself. Studies on adaptive signal control emphasize the benefits for traffic throughput and delay reduction, but few directly extend pedestrian crossing time based on real-time trajectory data. Moreover, vulnerable road users (VRUs), such as wheelchair users or visually impaired pedestrians, remain underrepresented in such studies [5, 6, 20, 27]. To address this gap, we introduce the following contributions:

(1) **NPLB: An adaptive pedestrian-safety system.** A real-time traffic signal architecture (Fig 1) that integrates a fine-tuned VRU YOLOv12 detector with an adaptive signal controller to dynamically extend pedestrian crossing times based on the detection of VRUs.
(2) **Comprehensive benchmarking of VRU detection models.** We conduct a comparative evaluation of state-of-the-art object detectors (YOLOv5, YOLOv11, YOLOv12, SSDLite, Faster R-CNN) on the BGVP dataset, establishing performance baselines for VRU detection.
(3) **State-of-the-art YOLOv12 model for VRU detection.** We fine-tune YOLOv12 for VRU detection and achieve an mAP@0.5 of 0.756, outperforming all tested models and demonstrating its suitability for real-time integration within NPLB.

3 NPLB (No Pedestrian Left Behind) System

Our research explores an approach to improving pedestrian safety through computer vision technology. We focus on enhancing detection capabilities specifically for vulnerable road users and implement proper mitigation through our NPLB system.

The NPLB system addresses a critical gap in pedestrian safety by dynamically adapting traffic signal timing based on real-time detection of vulnerable road users. The system architecture, illustrated in Figure 1, consists of three integrated layers: perception, detection, and control.

3.1 System Architecture

(1) **Perception Layer** processes video input through our fine-tuned YOLOv12 model, which detects VRUs and generates bounding boxes with class labels for each frame. These detections are then passed to `ByteTrack`, a multi-object tracking algorithm that maintains consistent track IDs across frames, enabling the system to follow individual pedestrians throughout their crossing journey.

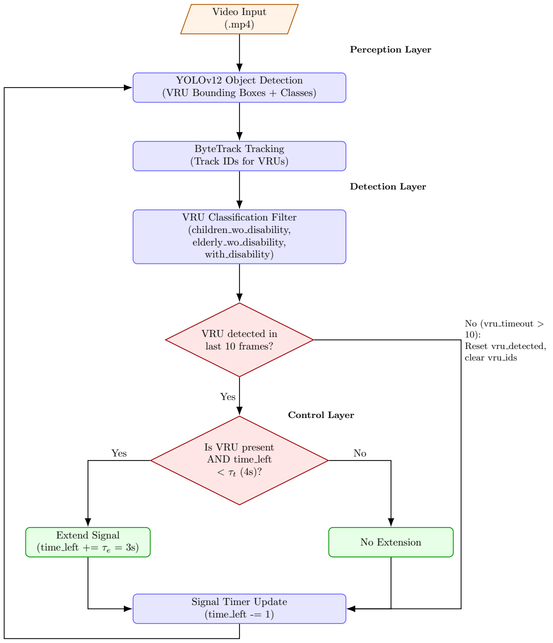

Figure 1: NPLB Architecture

(2) **Detection Layer** filters the tracked objects to identify VRUs specifically, focusing on three vulnerability categories: children without disabilities, elderly individuals without disabilities, and persons with disabilities. This layer maintains a set of unique VRU track IDs and implements a timeout mechanism to handle temporary occlusions or detection gaps. If no VRU is detected for 10 consecutive frames (approximately 0.33 seconds at 30 fps), the system resets its VRU tracking state to prevent false positives from stale detections.
(3) **Control Layer** implements the adaptive signal timing logic. When a VRU is detected in the crosswalk and the remaining signal time falls below a τ_t critical threshold (4 seconds), the system automatically extends the pedestrian phase by a τ_e predefined extension time (3 seconds). This threshold is based on typical walking speeds and crosswalk widths, ensuring that pedestrians who have already committed to crossing have sufficient time to complete their journey safely.

3.2 Assumptions and Scope

The current NPLB implementation assumes

(1) Pedestrians are crossing legally at designated crosswalks during pedestrian signal phases
(2) Pedestrians are crossing during pedestrian signal phases
(3) An environment with pre-existing clearly marked crosswalks
(4) Adequate lighting conditions for computer vision processing.

Illegal crossings or jaywalking scenarios require different detection and response strategies and are outside the scope of this work.

The extension time parameter can be calibrated based on local intersection characteristics, including crosswalk width, typical VRU walking speeds, and traffic patterns. We use a τ_e extension of 3 seconds.

Algorithm 1 No Pedestrian Left Behind (NPLB)

```
Input: video_stream, initial_signal_time
Parameters: $\tau_e$ = 3.0s (extension_time), $\tau_t$ = 4.0s (threshold_time), $n_{max}$ = 2 (max_extensions)
model ← YOLOv12.load("weights.pt")
vru_ids ← $\emptyset$ {Set of unique VRU track IDs}
vru_detected ← False
vru_timeout ← 0
time_left ← initial_signal_time
extension_count ← 0
for each frame in video_stream.track(tracker="ByteTrack") do
    detections ← frame.boxes
    vru_detected ← False {Reset per frame}
    for each box in detections do
        cls ← box.class
        track_id ← box.id
        if cls $\in$ {children_wo_disability, elderly_wo_disability, with_disability} then
            vru_detected ← True
            if track_id $\neq -1$ and track_id $\notin$ vru_ids then
                vru_ids ← vru_ids $\cup$ {track_id}
            end if
        end if
    end for
    if not vru_detected then
        vru_timeout ← vru_timeout + 1
        if vru_timeout > 10 then
            vru_detected ← False
            vru_ids ← $\emptyset$
            vru_timeout ← 0
        end if
    else
        vru_timeout ← 0 {Reset timeout}
    end if
    if vru_detected and time_left < $\tau_t$ and extension_count < $n_{max}$ then
        time_left ← time_left + $\tau_e$
        extension_count ← extension_count + 1
        output "EXTEND_SIGNAL"
    end if
    time_left ← time_left - 1
end for
```

3.3 Operational Logic

Algorithm 1 describes the complete NPLB workflow. The system operates on a per-frame basis, processing each video frame through the YOLOv12 detector with ByteTrack integration. For each detected bounding box, the system checks whether the object belongs to one of the three VRU classes. When a VRU is identified, its track ID is added to the active VRU set, and the timeout counter is reset.

The decision to extend signal timing is governed by two conditions:

(1) At least one VRU must be actively detected in the current frame or have been detected within the last 10 frames
(2) The remaining signal time must be less than the critical threshold.

When both conditions are satisfied, the system issues an `EXTEND_SIGNAL` command and increments the remaining time by the extension duration.

This approach ensures that the system only extends signals when necessary, balancing pedestrian safety with traffic flow efficiency. The timeout mechanism prevents indefinite signal extensions due to detection artifacts or pedestrians who have already cleared the crosswalk, while the track ID maintenance allows the system to recognize returning or persistently present VRUs.

4 Experiment Design and Evaluation

4.1 NPLB Evaluation

To evaluate the safety impact of the NPLB system, we conducted Monte Carlo simulations comparing fixed-time signals against NPLB control. Our simulation framework models pedestrian crossing scenarios with varying crosswalk widths, and walking speeds based on pedestrian types.

4.1.1 Simulation Framework. We simulated 10,000 crossing scenarios using pedestrian walking speeds from U.S. Department of Transportation guidelines [21]: average adults (4.00 ft/s), wheelchair users (3.55 ft/s), and elderly adults (2.80 ft/s). Pedestrian type distribution was based on 2023-2024 U.S. Census data [8, 22]: 18% elderly, 1.6% wheelchair users, and 80.4% general adults. Consistent with recent city-wide measurements reported by Moran and Laefer [19], crosswalk lengths in our study fall within a range of 30 to 60 feet. Longer crossings exceeding 50–60 feet are associated with higher pedestrian-collision risk [19]

Each simulation tracks individual pedestrians crossing at realistic speeds with randomized entry delays (0-3 seconds) to model natural crossing behavior. A pedestrian is considered *stranded* if they remain in the crosswalk when the signal phase expires. Monte Carlo simulation has proven effective in pedestrian safety research [7, 13], allowing comprehensive exploration of parameter space and edge cases.

4.1.2 NPLB System Parameters. The NPLB system operates with the following parameters (Table 1):

Table 1: System Parameters

Symbol	Parameter	Value	Description
ρ	Detection Rate	0.756	mAP@0.5
τ_e	Extension Time	3.0 s	Extension duration
τ_t	Threshold Time	4.0 s	Time-remaining trigger
n_{max}	Max Extensions	2	Allowed per crossing
α	Systemic Failure	0.26	1 - recall

τ_e and τ_t parameters were selected through sensitivity analysis to balance pedestrian safety with minimal traffic flow disruption.

4.1.3 Baseline Comparison. For the fixed-time baseline, we calculated signal durations using the Colorado Department of Transportation formula [1]: signal time = (crossing distance / walking speed) + buffer, where buffer ≥ 2 seconds. We used a 5-second buffer, representing conservative engineering practices.

4.1.4 Parameter Optimization. We conducted a parameter sweep across extension times (3.0-6.0 s) and threshold times (3.0-6.0 s) to identify optimal values (Figure 2). The analysis revealed that τ_e = 3.0 seconds provides sufficient protection while minimizing signal extensions, and τ_t = 4.0 seconds offers the most significant stranding reduction without excessive impact on total signal duration. These values represent the minimum effective intervention that maintains safety benefits.

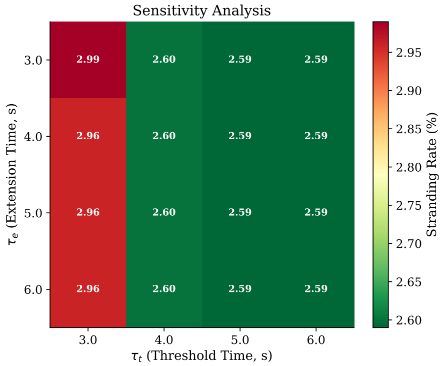

Figure 2: Heatmap of parameter sweep showing stranding rates across different combinations of extension time (τ_e) and threshold time (τ_t)

4.2 Model Evaluation

Mean average precision (mAP) [24] provides a meaningful evaluation over multiple classes for Computer Vision models. We describe it mathematically as shown in Equation 1:

$$mAP = \frac{1}{n} \cdot \sum_{i=1}^{n} AP_i \tag{1}$$

Intersection over Union (IoU) measures the overlap between a predicted bounding box and the ground-truth box, defined as the area of intersection divided by the area of their union. Higher IoU values indicate more accurate localization of the detected object.

mAP@0.5 considers a detection correct if the predicted bounding box overlaps the ground truth by at least 50% IoU, providing a lenient measure of detection. On the other hand, **mAP@[0.5:0.95]** averages the mean Average Precision over IoU thresholds from 0.5 to 0.95 in increments of 0.05, providing a stricter and more comprehensive evaluation that accounts for both detection and localization accuracy across multiple overlap levels.

4.3 Data Collection

After examining different pedestrian datasets including PIE [23], JAAD [16], and Caltech Pedestrians [11], we found that these datasets do not contain VRUs. We therefore selected the BGVP dataset [25] due to the availability of annotated images of VRUs.

The BGVP dataset follows the COCO [17] dataset format. To use the BGVP dataset with our YOLO models, we employ Algorithm 2 to convert the BGVP dataset format into YOLO format.

Algorithm 2 BGVP Dataset Preprocessing

```
Input: BGVP dataset with split archives and COCO annotations

Output: YOLO-formatted dataset
Create dataset configuration with 4 VRU classes
for each split in {train, validation, test} do
    Extract images from archive to organized directory structure
end for
for each split in {train, validation, test} do
    Load COCO annotations for split
    for each annotated image do
        Convert bounding boxes: YOLO format ⟶ COCO format
        Normalize coordinates by image dimensions
        Write YOLO label file with class and bbox per object
    end for
end for
Validate image and label counts match across all splits
return Preprocessed dataset ready for YOLO training
```

5 Model Implementation

To identify the most effective architecture for VRU detection, we evaluate 5 state-of-the-art object detection models fine-tuned on the BGVP dataset. Our evaluation includes three variants from the YOLO family (YOLOv5, YOLOv11, YOLOv12), known for real-time performance, alongside Faster R-CNN and SSDLite, which represent two-stage and lightweight detection approaches, respectively.

5.1 YOLOv5, YOLOv11, YOLOv12

For real-time VRU detection, we fine-tuned three YOLO object-detection models (v5, v11, v12) on the BGVP dataset. The YOLO models are single-stage detectors designed for speed and accuracy, and output bounding boxes, objectness scores, and class probabilities in a unified framework.

Training procedure. We initialized each model with pre-trained weights provided by `Ultralytics` and trained on the BGVP dataset with 3 VRU classes. Training parameters were set to ensure sufficient learning while mitigating overfitting:

- **Image size:** 640 × 640 pixels
- **Batch size:** 16
- **Epochs:** 500, with early stopping patience of 50 epochs
- **Data augmentation:** horizontal flips, mosaic augmentation, small geometric transformations (rotation ±10°, translation up to 10%, scaling up to 50%), and mild color augmentations (hue ± 0.015, saturation ± 0.7, brightness ± 0.4)

Implementation. Model training was performed using the `ultralytics` Python library. We monitor validation performance via the mean Average Precision (mAP) metric, saving the best weights automatically. Augmentation strategies, such as mosaic and horizontal flips, were included to improve generalization given the limited size of the VRU-specific dataset.

5.2 Faster R-CNN

For comparison with real-time YOLO models, we fine-tuned a Faster R-CNN model with a ResNet-50 backbone on the BGVP dataset. Faster R-CNN is a two-stage detector that first generates region proposals and then classifies each proposal while refining bounding box coordinates, offering high accuracy at the expense of inference speed.

Training procedure. The model is initialized with weights pre-trained on the COCO dataset and trained to detect the same 3 VRU classes. Key training parameters include:

- **Backbone:** ResNet-50 with Feature Pyramid Network (FPN)
- **Batch size:** 2
- **Epochs:** 100, with early stopping patience of 10 epochs
- **Optimizer:** Stochastic Gradient Descent (SGD) with learning rate 0.005, momentum 0.9, weight decay 0.0005
- **Learning rate scheduler:** StepLR with step size 3 and decay factor 0.1
- **Data augmentation:** Random horizontal flips (50%) and normalization

Implementation. Training is performed using PyTorch and the TorchVision library. We monitor validation performance using mean Average Precision (mAP) and save the best-performing model weights automatically. Faster R-CNN is computationally more expensive than YOLO, so fewer epochs are used to balance training time and accuracy.

5.3 SSDLite

To explore lightweight detection architectures suitable for mobile and edge deployment, we fine-tuned an SSDLite320 model with a MobileNetV3-Large backbone on the BGVP dataset. SSDLite is a single-stage detector that uses depthwise separable convolutions to significantly reduce computational cost while maintaining competitive accuracy, making it well-suited for resource-constrained environments.

Training procedure. The model is initialized with weights pre-trained on the COCO dataset and fine-tuned to detect 3 VRU classes plus background (5 classes total). We replaced the original classification head with a new `SSDLiteClassificationHead` configured for our target classes while preserving the backbone and feature extraction layers. Key training parameters include:

- **Input resolution:** 320 × 320 pixels
- **Backbone:** MobileNetV3-Large with depthwise separable convolutions
- **Batch size:** 2
- **Epochs:** 150, with early stopping patience of 30 epochs
- **Optimizer:** AdamW with learning rate 0.001 and weight decay 0.0005
- **Learning rate scheduler:** StepLR with step size 3 and decay factor 0.1

Implementation. Training is performed using `PyTorch` and `TorchVision's` detection module. We monitor validation performance using mean Average Precision (mAP). Category IDs from the COCO-format annotations are remapped to 1-indexed labels to ensure compatibility with the model's classification head, where background is implicitly class 0.

6 Results

6.1 NPLB

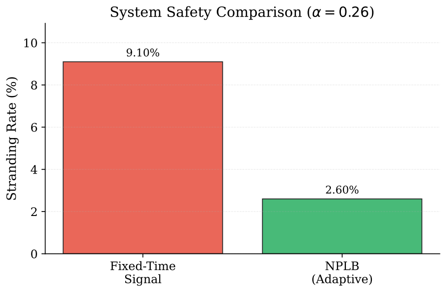

Figure 3: Comparison of stranding rates between fixed-time signal and NPLB adaptive control

Figure 3 demonstrates that NPLB achieved a 71.4% improvement in stranding rates compared to fixed-time signals — reducing stranding rates from 9.10% to 2.60%.

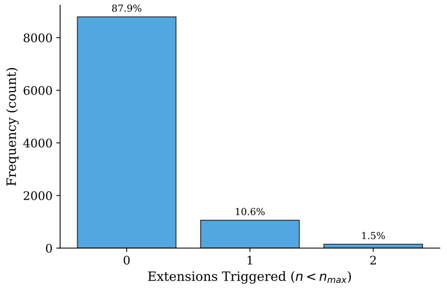

Figure 4: Frequency distribution of signal extensions triggered by NPLB system

The distribution of signal extensions (Figure 4) shows that 87.9% of crossing cycles require no intervention, because extensions are triggered only when VRUs are detected near signal expiration. NPLB did not impact the majority of crossing cycles, mostly granting 1 extension in 10.6% of crossing cycles and rarely 2 extensions in 1.5%.

The signal duration analysis (Figure 5) demonstrates that NPLB maintains the average signal time achieved by fixed-time signals, the upper bound for fixed-time signals was 22 seconds, while NPLB was 28 seconds. The extension remains modest and only affects 12.1% of scenarios where VRUs are at risk.

6.2 Object detection

Table 2 presents the performance of all five object detection models on the BGVP dataset. Among the YOLO variants, YOLOv12 achieved

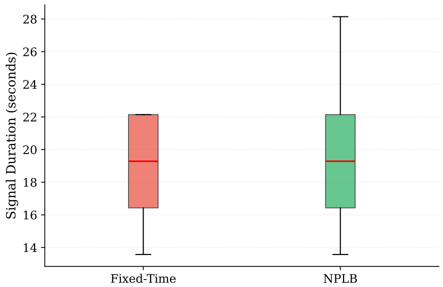

Figure 5: Box plot comparing pedestrian signal durations between fixed-time and NPLB systems

the highest mAP@0.5 of 0.756, followed closely by YOLOv5 (0.746) and YOLOv11 (0.741). When evaluated using the stricter mAP@[0.5:0.95] metric, which averages performance across multiple IoU thresholds, YOLOv11 slightly outperforms the other YOLO models with a score of 0.509, compared to 0.506 for YOLOv5 and 0.502 for YOLOv12.

The two-stage Faster R-CNN model achieved a mAP@0.5 of 0.659, performing below all YOLO variants despite its reputation for high accuracy. This suggests that the COCO pre-trained weights may require more extensive fine-tuning on the VRU-specific dataset. The lightweight SSDLite model obtained the lowest performance with a mAP@0.5 of 0.558, which is expected given its design prioritizes computational efficiency over accuracy.

Overall, the YOLO family demonstrates superior performance for VRU detection on the BGVP dataset, with all three variants achieving mAP@0.5 scores above 0.74 as shown in Figures 6, 7, and 8. The minimal performance differences between YOLOv5, YOLOv11, YOLOv12 suggest that any of these architectures would be suitable for deployment in the traffic signal control system.

Table 2: Comparison of Object Detection Models

Model	mAP@0.5	mAP@[0.5:0.95]
YOLOv12	0.756	0.502
YOLOv5	0.746	0.506
YOLOv11	0.741	0.509
Faster R-CNN	0.659	0.355
SSDLite (MobileNetV3)	0.558	0.329

7 Discussion and Conclusion

YOLO architectures perform better than other models described in Table 2 for the VRU detection task despite smaller size and faster inference. Therefore, Single-stage detectors like YOLO rank as competitive contestants against two-stage architectures in pedestrian safety applications. The 71.4% stranding reduction achieved by NPLB comes at little disruption to traffic flow, as seen in Figure 4. The key to this solution is selectivity: by triggering extensions only when VRUs approach signal expiration, the system intervenes rarely but effectively.

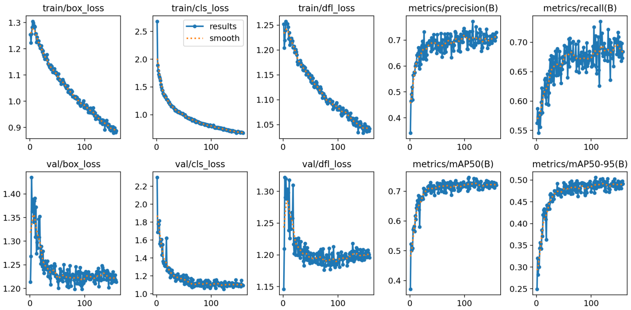

Figure 6: Training and validation metrics for YOLOv5 across 500 epochs, showing convergence of box loss, objectness loss, classification loss, precision, recall, and mAP.

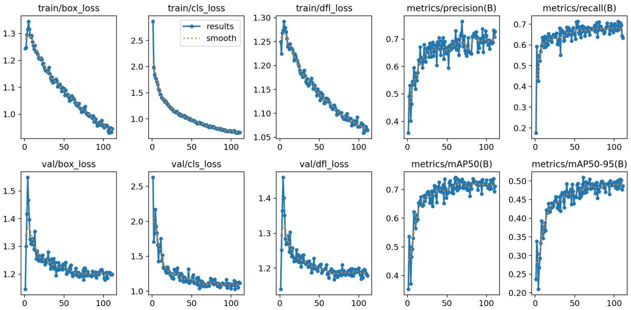

Figure 7: Training and validation metrics for YOLOv11 across 500 epochs, showing convergence of box loss, objectness loss, classification loss, precision, recall, and mAP.

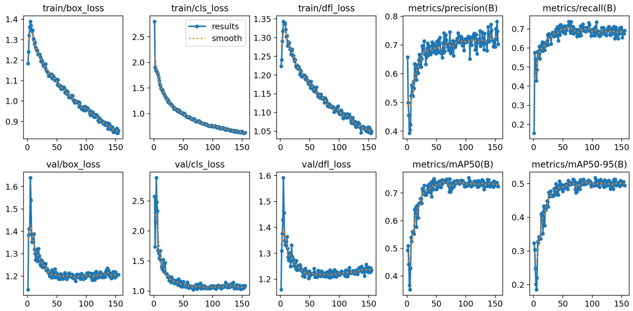

Figure 8: Training and validation metrics for YOLOv12 across 500 epochs, showing convergence of box loss, objectness loss, classification loss, precision, recall, and mAP.

Current transportation infrastructure may unintentionally favor able-bodied pedestrians. NPLB addresses this equity gap by accommodating Vulnerable Road Users. However, camera-based monitoring may raise privacy concerns for stakeholders — posing community engagement and transparent data policies as critical.

7.1 Limitations

Our Monte Carlo evaluation worked under idealized assumptions: 1) consistent lighting, 2) legal crossing behavior, and 3) all detected pedestrians are crossing. Real-world deployment will encounter visual occlusions caused by vehicles, weather conditions and edge cases that our simulations do not capture.

The fixed threshold parameters (τ_e, τ_t) may need to be adjusted based on intersection length, the demographics of pedestrians and the traffic patterns. A more robust production system could potentially learn thresholds adaptively from local data rather than relying on pre-defined constants.

References

[1] 2024. Traffic Documents and Publications. https://www.codot.gov/safety/traffic-safety/assets/documents

[2] Ahmed S. Abdelrahman, Mohamed Abdel-Aty, and Dongdong Wang. 2025. Video-to-Text Pedestrian Monitoring (VTPM): Leveraging Large Language Models for Privacy-Preserve Pedestrian Activity Monitoring at Intersections. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision Workshops (WACVW)*. IEEE, 329–338. https://doi.org/10.1109/WACVW65960.2025.00043

[3] Ahmed S. Abdelrahman, Seungil Ham, Mohamed A. Aty, et al. 2025. VRUCrossSafe for Crossing Intention Prediction of Vulnerable Road Users. *npj Intelligent Transportation* (2025). https://www.nature.com/articles/s44333-025-00037-5.

[4] Md Jaber Ahsan, Jiaqi Ma, et al. 2025. Evaluating the Safety Impact of Midblock Pedestrian Signals (MPS). *Accident Analysis & Prevention* (2025). https://pubmed.ncbi.nlm.nih.gov/39591735/.

[5] Gorkem Akyol, Ismet Goksad Erdagi, Mehmet Ali Silgu, and Hilmi Berk Celikoglu. 2020. Adaptive Signal Control to Enhance Effective Green Times for Pedestrians: A Case Study. *Transportation Research Procedia* 47 (2020), 704–711. https://doi.org/10.1016/j.trpro.2020.03.150 22nd EURO Working Group on Transportation Meeting, EWGT 2019, 18th – 20th September 2019, Barcelona, Spain.

[6] R. Avelar et al. 2021. Developing Crash Modification Factors for Adaptive Signal Control Technologies (ASCT). https://www.fhwa.dot.gov/publications/research/safety/20072/20072.pdf.

[7] Alben Rome Bagabaldo and Jürgen Hackl. 2025. Improving Pedestrian Safety at Intersections Using Probabilistic Models and Monte Carlo Simulations. *arXiv preprint arXiv:2503.07805* (2025).

[8] US Census Bureau. 2025. Older Adults Outnumber Children in 11 States and Nearly Half of U.S. Counties. https://www.census.gov/newsroom/press-releases/2025/older-adults-outnumber-children.html

[9] Manuel Burgos-Payán, Francisco-Javier Correa-Moreno, and Jesús-Manuel Riquelme-Santos. 2012. Improving the energy efficiency of street lighting. A case in the South of Spain. In *2012 9th International conference on the European energy market*. IEEE, 1–8.

[10] Jinkun Cao, Jiangmiao Pang, Xinshuo Weng, Rawal Khirodkar, and Kris Kitani. 2023. Observation-Centric SORT: Rethinking SORT for Robust Multi-Object Tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 9686–9696. https://openaccess.thecvf.com/content/CVPR2023/papers/Cao_Observation-Centric_SORT_Rethinking_SORT_for_Robust_Multi-Object_Tracking_CVPR_2023_paper.pdf

[11] Piotr Dollar, Christian Wojek, Bernt Schiele, and Pietro Perona. 2009. Pedestrian detection: A benchmark. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*. 304–311. https://doi.org/10.1109/CVPR.2009.5206631

[12] Divya Gangwani and Pranav Gangwani. 2021. Applications of machine learning and artificial intelligence in intelligent transportation system: A review. *Applications of Artificial Intelligence and Machine Learning: Select Proceedings of ICAAAIML 2020* (2021), 203–216.

[13] Zhi Huang, Xiangyi Liu, Xiaolin Song, and Yin He. 2017. Probabilistic risk assessment for pedestrian–vehicle collision considering uncertainties of pedestrian mobility. *Traffic Injury Prevention* 18, 6 (2017), 650–656. https://doi.org/10.1080/15389588.2017.1283026 arXiv:https://doi.org/10.1080/15389588.2017.1283026 PMID: 28112561.

[14] Abdoulaye Kama, Moussa Diallo, Mamadou S. Drame, Mamadou L. Ndiaye, Ababacar Ndiaye, and Pape A. Ndiaye. 2017. Monitoring the Performance of Solar Street Lights in Sahelian Environment: Case Study of Senegal. In *2017 10th International Conference on Developments in eSystems Engineering (DeSE)*. 56–61. https://doi.org/10.1109/DeSE.2017.43

[15] Marta Kolasa. 2016. The concept of intelligent system for street lighting control using artificial neural networks. *Prz. Elektrotechniczny* 92, 7 (2016), 32–37.

[16] Iuliia Kotseruba, Amir Rasouli, and John K. Tsotsos. 2016. Joint Attention in Autonomous Driving (JAAD) Dataset. https://arxiv.org/abs/1609.04741.

[17] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. 2015. Microsoft COCO: Common Objects in Context. arXiv:1405.0312 [cs.CV] https://arxiv.org/abs/1405.0312

[18] Prabu Mohandas, Jerline Sheebha Anni Dhanaraj, and Xiao-Zhi Gao. 2019. Artificial Neural Network based Smart and Energy Efficient Street Lighting System: A Case Study for Residential area in Hosur. *Sustainable Cities and Society* 48 (2019), 101499. https://doi.org/10.1016/j.scs.2019.101499

[19] Marcel E. Moran and Debra F. Laefer. 2024. Multiscale Analysis of Pedestrian Crossing Distance. *Journal of the American Planning Association* 0, 0 (2024), 1–15. https://doi.org/10.1080/01944363.2024.2394610 arXiv:https://doi.org/10.1080/01944363.2024.2394610

[20] D. Muley, S. Badi, M. Kharbeche, W. Alhajyaseen, and C. Dias. 2017. An Overview of Pedestrian Signal Setting and Design Practices. *Procedia Computer Science* 109 (2017), 326–333. https://doi.org/10.1016/j.procs.2017.05.357

[21] U.S. Department of Transportation. 2016. Lesson 8 - Federal Highway Administration University Course on Bicycle and Pedestrian Transportation, July 2006 - FHWA-HRT-05-100. https://www.fhwa.dot.gov/publications/research/safety/pedbike/05085/pptchapt8.cfm

[22] U.S. Department of Transportation. 2024. Secretary Buttigieg Announces Proposed Rule to Ensure Passengers Who Use Wheelchairs Can Fly with Dignity | US Department of Transportation. https://www.transportation.gov/briefing-room/secretary-buttigieg-announces-proposed-rule-ensure-passengers-who-use-wheelchairs-can

[23] Amir Rasouli, Iuliia Kotseruba, and John K. Tsotsos. 2019. PIE: A Large-Scale Dataset and Models for Pedestrian Intention Estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 6262–6271. https://openaccess.thecvf.com/content_ICCV_2019/papers/Rasouli_PIE_A_Large-Scale_Dataset_and_Models_for_Pedestrian_Intention_Estimation_ICCV_2019_paper.pdf

[24] Tobias Schlosser, Michael Friedrich, Trixy Meyer, and Danny Kowerko. 2024. A consolidated overview of evaluation and performance metrics for machine learning and computer vision. *Tobias Schlosser, Michael Friedrich, Trixy Meyer, and Danny Kowerko–Junior Professorship of Media Computing, Chemnitz University of Technology* 9107 (2024).

[25] Devansh Sharma, Tihitina Hade, and Qing Tian. 2022. Comparison Of Deep Object Detectors On A New Vulnerable Pedestrian Dataset. *arXiv preprint arXiv:2212.06218* (2022).

[26] Asis Kumar Tripathy, Alekha Kumar Mishra, and Tapan Kumar Das. 2017. Smart lighting: Intelligent and weather adaptive lighting in street lights using IOT. In *2017 International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT)*. 1236–1239. https://doi.org/10.1109/ICICICT1.2017.8342746

[27] Xinhai Wang and Wenhua Shao. 2025. Networked Sensor-Based Adaptive Traffic Signal Control for Dynamic Flow Optimization. *Sensors* 25, 11 (2025), 3501. https://doi.org/10.3390/s25113501

[28] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. 2017. Simple Online and Realtime Tracking with a Deep Association Metric. *arXiv preprint arXiv:1703.07402* (2017). https://arxiv.org/abs/1703.07402

[29] World Health Organization. 2023. Pedestrian Safety: A Road Safety Manual for Decision-Makers and Practitioners, 2nd Edition. https://www.who.int/publications/i/item/9789240072497. Accessed 2025-09-18.

[30] Yifu Zhang, Peize Sun, Yi Jiang, Dongdong Yu, Fucheng Weng, Zehuan Yuan, Ping Luo, Wenyu Liu, and Xinggang Wang. 2022. Bytetrack: Multi-object tracking by associating every detection box. In *European conference on computer vision*. Springer, 1–21.