Lightweight Day and Night Pedestrian Detection for Sensitive Driver Assistance

Alexandru-Rafael Hodor
University of Craiova
Craiova, Romania
halexandrurafael@gmail.com

Paul-Stefan Popescu
University of Craiova
Craiova, Romania
stefan.popescu@edu.ucv.ro

Marian Cristian Mihaescu
University of Craiova
Craiova, Romania
cristian.mihaescu@edu.ucv.ro

ABSTRACT

Pedestrians account for a disproportionate share of urban traffic casualties, most of which occur at speeds below 50 km/h, where timely driver warning could avert impact. We present a low-cost, camera-only Advanced Driver-Assistance System that detects, tracks, and range-estimates pedestrians in real time on commodity hardware. The core detector is YOLO11n[1] (≈ 4 M parameters) fine-tuned on a 50-50 day-night subset of the EuroCity Persons dataset[2] (4 650 day and 4,222-night training images), eliminating the daylight bias typical of existing models. The system sustains a latency of 33 ms per 640 × 480 frame (<1 GB RAM). A ByteTrack module maintains identities across frames, and a pin-hole projection coupled with a time-to-collision filter triggers visual alerts when braking distance is insufficient. Unlike prior work that assumes discrete GPUs, multispectral sensors or daylight scenes, our pipeline delivers illumination-robust detection, constant-frame-rate inference and modular open-source implementation on hardware already present in budget vehicles. The results demonstrate that adequate pedestrian warning can be achieved without specialised sensors or expensive accelerometers, paving the way for wider deployment in cost-sensitive markets and retrofit scenarios.

Author Keywords
Machine learning; object detection; YOLO; ECP; pedestrians

ACM Classification Keywords
H.5.m Information Interfaces and Presentation (e.g., HCI): Miscellaneous; I.4.8 Artificial Intelligence: Scene Analysis.

General Terms
Human Factors; Design; Measurement.

DOI: 10.37789/icusi.2025.26

INTRODUCTION

Road traffic collisions remain a leading cause of accidental death worldwide, and the burden is felt most acutely by pedestrians, the least protected road users. Whereas vehicle occupants benefit from crumple zones, airbags, and seat belts, a pedestrian struck even at moderate speed faces a high risk of severe injury or fatality. Recent analyses of advanced-driver-assistance systems (ADAS) underline the scale of the preventable problem: if just the six most common ADAS functions were deployed fleet-wide, overall crash frequency could fall by almost a quarter, with automatic emergency braking alone reducing pedestrian impacts by 28 % in the United Kingdom, or roughly 19,000 avoided crashes each year. These statistics frame the overarching motivation of this work, augmenting human drivers with computer-vision capabilities that do not tire, blink, or become distracted.

Pedestrian detection by on-board cameras is a cornerstone of such capabilities. Yet, it remains challenging in the very scenarios where human vision struggles most: low illumination, glare, and cluttered urban backgrounds. Traditional benchmark datasets, such as Caltech[3] or KITTI[4], are dominated by daytime footage, so detectors trained on them often suffer from a daylight bias and degrade sharply after dusk. To address this gap, the EuroCity Persons (ECP) dataset provides more than 200,000 annotated pedestrians captured across 31 European cities, balanced across seasons, weather conditions, and—crucially—day and nighttime scenes. Its high-resolution images preserve the fine detail needed for recognising distant or partially occluded pedestrians in complex traffic. Leveraging ECP therefore, promises not only higher accuracy but also greater robustness to illumination changes.

On the algorithmic side, one-stage detectors of the You Only Look Once (YOLO) family have become de facto standards for high-frame-rate perception. By regressing bounding boxes and class scores in a single forward pass, YOLO avoids the multi-stage region-proposal bottleneck of earlier R-CNN variants and consistently achieves real-time throughput on commodity GPUs. Lightweight iterations—most recently YOLO11n—compress model depth and width while retaining the decoupled head architecture that improves small-object recall, making them attractive for edge devices such as in-vehicle integrated graphics processors. Nevertheless, a generic model trained on the MS-COCO benchmark is sub-optimal for monoclass pedestrian detection and cannot exploit ECP's night-time diversity.

This paper presents a streamlined pipeline that bridges that gap. First, we curate a balanced ECP subset containing 4 650

[1] https://docs.ultralytics.com/models/yolo11/

[2] https://eurocity-dataset.tudelft.nl

[3] https://data.caltech.edu/records/f6rph-90m20

[4] https://www.cvlibs.net/datasets/kitti/

daytime and 4 222 night-time images for training and an equal 770 (for day) and 770 (for night) for validation, maintaining a near-50/50 illumination ratio to mitigate bias. Second, we fine-tune YOLO11n on this subset, preserving the small model footprint (≈4 M parameters). Third, we export the network to ONNX and run inference via DirectML on an AMD Ryzen iGPU, achieving ≈30 ms per 640 × 480 frame while consuming <350 MB of system memory . A simple pin-hole projection estimates pedestrian range, and a heuristically gated time-to-collision filter triggers a visual alert in a Tkinter graphical interface, demonstrating an end-to-end assistive prototype that operates without dedicated GPU hardware.

RELATED WORK

Pedestrian detection is a relevant task as stated in [1] which provides a comprehensive survey of deep-learning approaches to pedestrian detection for autonomous driving, highlighting three persistent bottlenecks—occlusion, illumination variation (especially night-time scenes), and the small-object problem—and reviewing how modern one-stage detectors (YOLOv3/v4), two-stage models (Faster R-CNN), and lightweight backbones (MobileNet-SSD) attempt to mitigate them. The authors catalogue more than 40 recent studies, compare performance on canonical benchmarks (Caltech, CityPersons, KITTI, NightOwls[5]) and emphasise that data imbalance toward daylight imagery remains a root cause of poor nocturnal recall. They further argue that real-time deployment on embedded hardware is still constrained by model size and compute budgets, noting a trade-off between speed-oriented "nano" variants and the accuracy gains brought by attention or multi-spectral fusion modules. Finally, the survey calls for curated day-/night-balanced datasets and task-specific compression techniques as key directions for future research. Our work answers both of these open recommendations: we construct a 50/50 day-night subset of EuroCityPersons to tackle the illumination bias, and we fine-tune the 4 M-parameter YOLO11n architecture, achieving state-of-the-art on that subset while sustaining 30 ms inference on an integrated GPU thus extending the lines of inquiry pinpointed by authors of [1].

Liu et al. (2024) [2] tackle the twin obstacles of scale variance and crowding at signalised intersections by introducing YOLOv8-CB, a compact derivative of YOLO-v8n that layers three architectural tweaks: a lightweight Cascade-Fusion Network (CFNet) to enrich deep-shallow feature interplay, a CBAM attention module to highlight salient spatial–channel cues, and a bidirectional BIFPN path for weighted multi-resolution fusion. Tested on dense-traffic footage, the model lifts detection accuracy by +2.4 points while trimming parameters (-6.45 %) and FLOPs (-6.74 %), clocking 10.8 ms per 640 × 640 frame—evidence that judicious feature re-use can surpass the vanilla YOLO-v8n speed/accuracy frontier without ballooning compute. The authors nonetheless leave the illumination-robustness question open, as their experiments focus on daylight scenes; moreover, the reliance on GPU-class hardware contrasts with our DirectML-enabled iGPU deployment.

Raza et al. (2023) [3] confront the nighttime surveillance setting with a classical machine-learning pipeline that forgoes deep nets altogether: after background subtraction on infrared (IR) video, a random forest classifier segments pedestrian silhouettes, multiple template matching locates candidate bounding boxes, and a particle filter data association scheme maintains tracks across frames. On their proprietary IR dataset the authors report 93 % segmentation accuracy, 90 % detection accuracy, and 81 % multi-target tracking accuracy, crediting the combination of handcrafted features and pixel-level verification for robustness under poor illumination. While the work demonstrates that low-cost IR sensors can mitigate the lighting issue, its reliance on template matching limits scalability to crowded scenes, and the absence of GPU-friendly deep learning precludes real-time deployment on embedded ADAS hardware. Moreover, because the camera is static and the field of view restricted to fixed surveillance corridors, the method does not address ego-motion, range estimation, or the wide range of pedestrian poses encountered in forward-looking automotive footage—gaps our balanced EuroCityPersons training protocol and YOLO11n optimisation explicitly tackle.

In paper [4], Li et al. (2023) pursue the speed-accuracy sweet-spot by re-engineering YOLOv5s into YOLOv5s-GAM/Ghost, a 4.9 M-parameter variant that swaps standard convolutions for Ghost/GhostC3 blocks, injects a Global Attention Mechanism (GAM) between neck and head, and replaces the GIoU loss with a −CIoU term better aligned with bounding-box regression Evaluated on the WiderPerson benchmark, the design trims 13.2 % FLOPs while nudging mean average precision upward by +1.0 points, showing that judicious channel-redundancy pruning and attention injection can offset the accuracy typically lost to lightweighting. Although the authors acknowledge illumination challenges, their experiments focus exclusively on daylight RGB data and rely on a discrete GPU for real-time throughput. Our work complements these gaps by balancing day- and night-time examples in EuroCityPersons and demonstrating sub-35 ms inference on an integrated GPU.

Hsu and Yang (2023) [5] attack the chronic "blurry-CCTV" corner-case by front-loading the detector with a Multi-scale Structure-Enhanced Super-Resolution (MsSE-SR) pre-processor. Using a stationary-wavelet transform, each low-resolution (LR) frame is split into frequency-specific sub-images; dedicated branches reconstruct high- and low-frequency details and then exchange cues through a high-to-low sub-network information-transfer (H2LSnIT) module, yielding a sharper, structure-faithful up-scaled image.

[5] https://www.nightowls-dataset.org

Coupled with a YOLOv4 head, the end-to-end pipeline significantly improves pedestrian average precision on LR benchmarks, demonstrating that optical-quality restoration can be as impactful as detector redesign when cameras are the bottleneck. However, the study limits itself to daylight RGB footage and assumes GPU-class compute, leaving illumination robustness and true edge-device deployment open—gaps our day/night-balanced EuroCityPersons fine-tuning and DirectML iGPU runtime explicitly address

Zhang et al. (2024) push pedestrian perception [6] into the multi-sensor era with MMPedestron, a "generalist" transformer that ingests any combination of RGB, infrared, depth, LiDAR or event-camera frames through a single unified encoder. Two learnable tokens, MAA and MAF, adaptively fuse modality-aware and modality-agnostic representations before a shared detection head, letting the network swap seamlessly between sensor pairs at test time. To train and benchmark such flexibility they aggregate existing corpora and introduce MMPD, the first large-scale multimodal pedestrian dataset, adding a new event-camera subset (EventPed). Joint training lifts performance to 71.1 AP on COCO-Persons [7] and 72.6 AP on LLVIP, while remaining 30 × smaller than InternImage-H on CrowdHuman—evidence that a single compact backbone can rival specialist models across modalities. Nevertheless, the study presumes discrete-GPU resources and does not dissect illumination bias within the RGB stream; by contrast, our work zeroes in on a lone, low-cost RGB sensor, balances day- and night-time footage in EuroCityPersons [9], and validates real-time inference on an integrated GPU.

Last paper discussed in related work is also a survey [10] as the first one because Ghari et al. (2024) deliver the first comprehensive survey devoted exclusively to low-light pedestrian detection, reviewing more than 150 studies across classical vision, deep RGB, multispectral fusion, and radar–vision hybrids. Their meta-analysis exposes three quantitative gaps: (i) dataset bias—nearly half of all papers benchmark solely on KAIST, with real-world night-driving videos used in < 6 % of works; (ii) methodological skew toward early- or halfway-fusion CNNs that merge visible and FIR channels, leaving single-sensor RGB solutions under-explored; and (iii) deployment neglect, as only 9 % of surveyed methods report latency or energy figures pertinent to embedded ADAS. The authors conclude that illumination-balanced datasets and compute-efficient monocular detectors are required for wider adoption. Our study responds directly: we craft a 50/50 day-night subset of EuroCityPersons and fine-tune a 4 M-parameter YOLO11n that clears 30 ms per 640 p frame on an integrated GPU—thereby addressing both the data-imbalance and edge-inference shortcomings highlighted by Ghari et al.

PROPOSED APPROACH

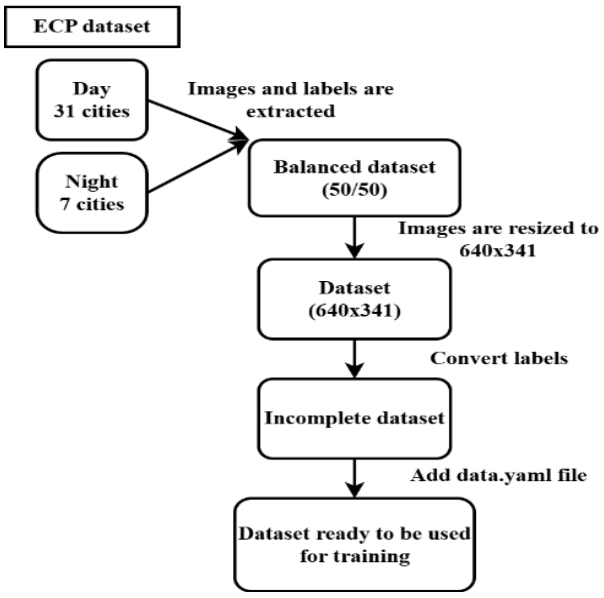

Figure 1. Dataset preparation

Figure 1 presents the preprocessing of the dataset. Since the same model would be used for varying light conditions, a 50/50 distribution for day and night images was aimed for. All night images were extracted, along with 150 day images per city for the training set and 25 day images per city for the validation set. This resulted in a training split of 4650 day images and 4222 night images (52.41% day / 47.59% night), and a balanced validation split of 770 day images and 770 night images. The training-to-validation ratio achieved was 85.2%/14.8%. The images were also resized to a maximum resolution of 640 pixels, as the model uses 640x640 as an input and it also contributes to faster training times. All labels were converted to .txt YOLO format, and all classes except pedestrians were removed. The data.yaml file was also added and the dataset was ready to be used for training.

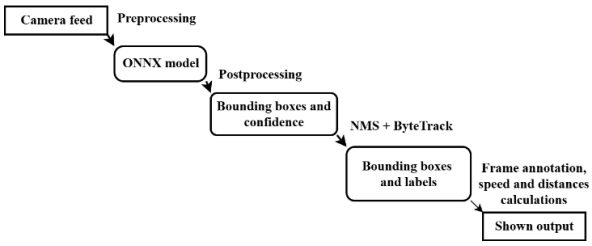

Figure 2. Data flow diagram

The fine-tuned model will perform real-time inference using the ONNX Runtime. This runtime was chosen for its support of DirectML, a hardware-accelerated DirectX 12 API from Microsoft, enabling the model's deployment even on integrated GPUs commonly found in lower-cost or embedded devices, thereby increasing system accessibility.

As can be observed in Figure 2, the system aims for a simple design to facilitate its implementation in a vehicle. For this reason, it is necessary to approximate the speed at which each pedestrian approaches the car. For this, each pedestrian will be tracked from frame to frame using Supervision's implementation of ByteTrack. The pinhole camera model will be employed to ascertain the approximate distance between the camera and the pedestrian (necessitating the use of an average height for this calculation). Once these distances are determined, they can be stored in a vector, and the approaching speed can be calculated over an average of 5 frames. This speed is derived from the rate at which distances change between frames, combined with the time elapsed between those frames, yielding an approximate speed. Based on this calculated speed and the maximum braking G-force provided by the user via the application, the driver can be alerted. This alert is visually conveyed by turning the bounding boxes red if a pedestrian enters a critical zone (i.e., too close for the driver to brake in time).

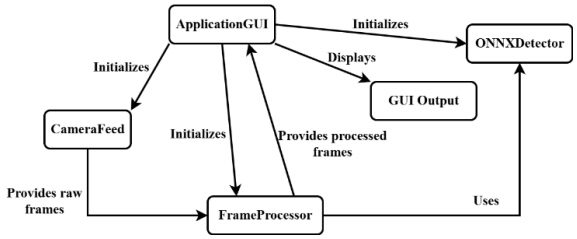

Figure 3. System architecture diagram

As shown in Figure 3, the architecture of the system adopts a modular design. The system is built using four components: the CameraFeed, dedicated to acquiring raw video frames; the ONNXDetector, which handles hardware-accelerated pedestrian detection and tracking; the FrameProcessor, responsible for real-time safety calculations and frame annotation; and the ApplicationGUI, which functions as both the user interface and central control hub.

The ApplicationGUI controls the entire system by initializing the CameraFeed, ONNXDetector, and FrameProcessor. From there, the CameraFeed continuously supplies raw frames to the FrameProcessor. The FrameProcessor then uses the ONNXDetector for object detection and tracking. It performs crucial safety calculations, including distance, speed, and required braking distance, before annotating the frames. Then these processed frames and various performance metrics are passed to the ApplicationGUI for them to be displayed. This decoupled structure facilitates efficient data flow and ensures a clear division of responsibilities across the modules.

EXPERIMENTAL RESULTS

The EuroCity Persons (ECP) dataset is a valuable resource for pedestrian detection, especially in autonomous driving and driver-assistance systems. What makes it special is its diversity and scale, the dataset includes images from 31 cities across 12 European countries, captured in all four seasons. It contains 40217 daytime images (with 183004 annotated pedestrians) and 7118 nighttime images (with 35309 annotations), all in high resolution. The dataset also covers different weather conditions, including both dry and wet.

One limitation is that ECP only includes European data, meaning models trained on it might not generalize as well to other regions where infrastructure, or visual conditions differ. Even so, given its wide variety of European environments and the sheer number of annotations, ECP remains a highly relevant benchmark for developing and testing pedestrian detection systems in real-world European applications.

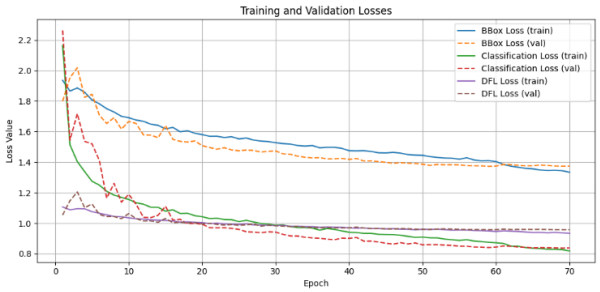

Figure 4. Training and Validation losses

As can be seen in Figure 4 the training is progressing very fast in the first 10-15 epochs evidenced by a steep decrease in all the training and validation loss components. After this we can observe that the rate of training significantly slows down. Furthermore, we can observe that the training is approaching completion and the model is generalizing effectively because the respective training and validation loss curves for each component converge and flatten out, indicating that the model is no longer making significant improvements and is stable on both seen and unseen data, without signs of overfitting.

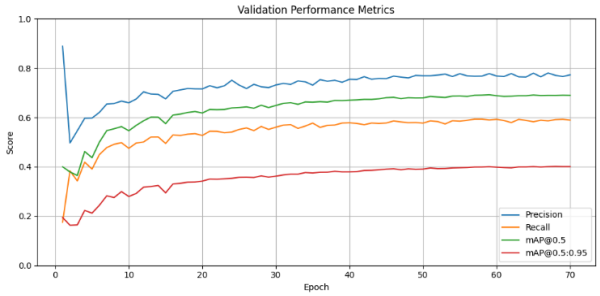

Figure 5. Validation performance metrics

The validation metrics exhibit trends inversely related to the training and validation losses during the model's learning process. As can be observed in Figure 5, a noticeable improvement in the performance metrics is evident during the initial 10 to 15 epochs. Following this rapid progress, the rate of improvement significantly slows down, and the metrics mostly stabilize after epoch 50, suggesting that the

model is performing at its best under the current training circumstances.

Additionally, the achieved pedestrian detection $mAP_{@0.5:0.95}$ of 0.40 on the ECP dataset is remarkably close to the $mAP_{@0.5:0.95}$ value of 0.395 reported by Ultralytics [8] for this model on the COCO[6] dataset. We can also observe that the mAP values don't start from 0, which is likely attributed to the model being trained on the COCO dataset which contains a person class that provides a solid base for fine-tuning for pedestrians.

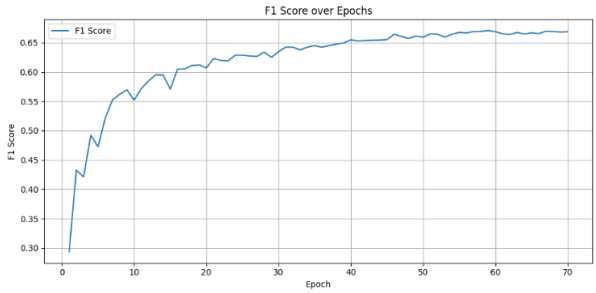

Figure 6. F1 score over epochs

The F1 score from Figure 6 shows similar tendencies to the other performance metrics, the values improve considerably in the first 10-20 epochs, after which the rate of improvement slows down. From epoch 50 to epoch 70 a plateau can be observed, indicating that the model has been successfully trained.

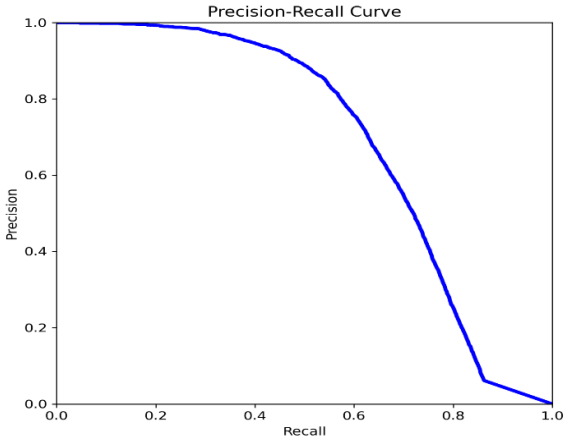

Figure 7. Precision-Recall curve

The obtained Precision-Recall (PR) curve, as shown in Figure 7, demonstrates a very high precision (close to 1) for low recall values (until 0.4). This indicates that with a strict classification threshold, the model makes very few correct positive predictions, resulting in high precision, though at reduced recall.

As the decision threshold relaxes, recall increases while precision gradually decreases until a recall of approximately 0.5-0.6. This trade-off signifies identifying more positive examples at the cost of increased false positives.

A notable characteristic is the sharp drop in precision from 0.8-0.9 to low values (0.1-0.2) around a recall of 0.6-0.8. This decline suggests that at very high recall numbers, the model generates many wrong positive predictions, drastically reducing precision.

In conclusion, the PR curve highlights an efficient compromise region in the recall range of 0.2-0.5. A decision threshold around 0.4 is identified as a functional equilibrium point between precision and recall, suitable for the application's requirements.

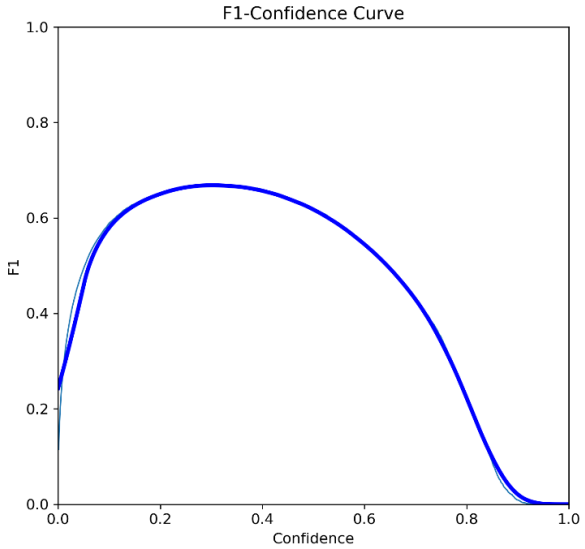

Figure 8. F1-Confidence curve

Based on Figure 8, the optimal confidence number for the model can be easily approximated. The F1 score, which balances precision and recall, rises significantly as the confidence threshold increases from very low values. It reaches its peak performance, approximately 0.65, at a confidence threshold of 0.35. Beyond this point, increasing the confidence threshold further results in a gradual decline in the F1 score, as the model becomes overly stringent and misses too many true positives. Consequently, a confidence threshold of 0.35 represents the optimal operating point for this model, effectively balancing its capacity to accurately identify pedestrians with minimizing both false positives and false negatives.

Additionally, the system's potential for mobile deployment was examined. Testing involved running the model on an Android device (a Samsung S21 Exynos smartphone was

[6] https://cocodataset.org/#home

used for testing), utilizing the ONNX Mobile Runtime. However, preliminary evaluations revealed sub-optimal performance, with inference times consistently ranging between 300-400ms per frame, most likely caused by the high resolution of the model. Given the real-time requirement of the application, this performance bottleneck proved to be a significant roadblock to further development within the scope of this paper. An example of the model running can be seen in Figure 9. Example of output. The corresponding code is available in the GitHub repository.

Figure 9. Example of output

The model exhibits strong real-time capabilities in the context of the system's performance, as illustrated by a representative output image in Figure 9. When running on an iGPU from an AMD Ryzen 5 5500U laptop, inference times consistently fall within a highly efficient range of 30-40 ms per frame. Compared to the previously discussed mobile implementation, this performance is approximately ten times faster, resulting in a fast frame rate that is essential for delivering real-time driver warnings. This speed is matched by the model's ability to discriminate between pedestrians, even in densely populated areas, accurately. Accurate distance estimation, a crucial part of the warning system, is made possible by consistently precise bounding boxes.

For reproducibility considerations all the code and diagrams are available on GitHub[7].

CONCLUSIONS

This research successfully developed and implemented a real-time pedestrian detection and warning system intended to enhance road safety. The system exhibits strong operational performance, running at a fast frame rate that is necessary to provide driver warnings instantly. The model demonstrates the ability to differentiate pedestrians, even in highly populated areas, producing consistently precise bounding boxes that facilitate critical distance estimation, a crucial part of the warning system.

Some notable features include its detection capabilities, which result from fine-tuning on the large and varied ECP dataset; its modular and maintainable architecture; and its operational performance on widely available hardware. The dataset's exclusive European contexts present a limitation that affects the model's ability to generalize to other parts of the world.

Preliminary testing on Android devices revealed subpar performance, which is a hurdle for broad portability without additional optimization. Future work will focus on expanding the model's training to a range of non-European driving conditions using additional datasets, testing optimization techniques to enhance performance across various platforms, including mobile devices, and potentially incorporating new sensor inputs, such as lidar or stereo cameras, to improve distance approximation accuracy and provide broader applicability. This study demonstrates the potential for developing and deploying real-time pedestrian warning systems, a promising method for enhancing road safety globally, regardless of the type of vehicle.

[7] https://github.com/Wisuy/Pedestrian-detection

REFERENCES

1. Iftikhar, S., Zhang, Z., Asim, M., Muthanna, A., Koucheryavy, A., & Abd El-Latif, A. A. (2022). Deep learning-based pedestrian detection in autonomous vehicles: Substantial issues and challenges. Electronics, 11(21), 3551.
2. Liu, Q., Ye, H., Wang, S., & Xu, Z. (2024). YOLOv8-CB: dense pedestrian detection algorithm based on in-vehicle camera. Electronics, 13(1), 236..
3. Raza, A., Chelloug, S. A., Alatiyyah, M. H., Jalal, A., & Park, J. (2023, January). Multiple pedestrian detection and tracking in night vision surveillance systems. In CMC (Vol. 75, pp. 3275-3289).
4. Li, M. L., Sun, G. B., & Yu, J. X. (2023). A pedestrian detection network model based on improved YOLOv5. Entropy, 25(2), 381..
5. Hsu, W. Y., & Yang, P. Y. (2023). Pedestrian detection using multi-scale structure-enhanced super-resolution. IEEE Transactions on Intelligent Transportation Systems, 24(11), 12312-12322..
6. Zhang, Y., Zeng, W., Jin, S., Qian, C., Luo, P., & Liu, W. (2024, September). When pedestrian detection meets multi-modal learning: Generalist model and benchmark dataset. In European Conference on Computer Vision (pp. 430-448). Cham: Springer Nature Switzerland..
7. Sun, C., Ai, Y., Qi, X., Wang, S., & Zhang, W. (2022). A single-shot model for traffic-related pedestrian detection. Pattern Analysis and Applications, 25(4), 853-865.
8. Jocher, G., Chaurasia, A., Stoken, A., Borovec, J., Kwon, Y., Michael, K., ... & Mammana, L. (2022). ultralytics/yolov5: v6. 2-yolov5 classification models, apple m1, reproducibility, clearml and deci. ai integrations. Zenodo.

9. Braun, M., Krebs, S., Flohr, F., & Gavrila, D. M. (2018). The eurocity persons dataset: A novel benchmark for object detection. arXiv preprint arXiv:1805.07193.

10. Ghari, B., Tourani, A., Shahbahrami, A., & Gaydadjiev, G. (2024). Pedestrian detection in low-light conditions: A comprehensive survey. Image and Vision Computing, 148, 105106.