Computer Vision and Image Understanding 269 (2026) 104788

Contents lists available at ScienceDirect

Computer Vision and Image Understanding

journal homepage: www.elsevier.com/locate/cviu

# TrackAid-DT: An egocentric vision dataset and benchmarks for lane detection in visually impaired athlete guidance☆

Gagan Narang [a], Alessandro Galdelli [a,*], Oleksandr Kuznetsov [b,c], Primo Zingaretti [a], Adriano Mancini [a]

[a] Department of Information Engineering, Università Politecnica delle Marche, Ancona, 60131, Italy
[b] Department of Theoretical and Applied Sciences, eCampus University, Novedrate, 22060, Italy
[c] Department of Intelligent Software Systems and Technologies, School of Computer Science and Artificial Intelligence, V.N. Karazin Kharkiv National University, Kharkiv, 61022, Ukraine

ARTICLE INFO

Dataset link: https://github.com/vrai-group/TrackAid-DT

*Keywords:*
Assistive computer vision
Lane detection
Semantic segmentation
Egocentric vision
Model quantization

ABSTRACT

Running on marked tracks is central to athletic training, still athletes with visual impairment depend on external guides limiting independence in training and competition. Advancements in assistive and navigation technologies offer new possibilities for inclusive sports environments. Therefore, there is an urgent need to develop inclusive mobility solutions that empower visually impaired athletes. In this work, we focus on an autonomous lane guidance system designed to assist athletes in independent navigation. To address this investigation, we propose an end-to-end vision-based system built upon our TrackAid Dataset (TrackAid-DT). The system operates under real-world conditions and supports safe athletic running by delivering vibrotactile haptic feedback for real-time guidance, in contrast to prior works that primarily address walking or jogging. TrackAid-DT comprises egocentric, pixel-annotated monocular images acquired from chest-mounted cameras under diverse orientations, speeds, and lighting conditions, and is designed to train segmentation models that enable the proposed guidance system and its deployment on edge devices. We benchmark multiple advanced segmentation models using eight-fold cross validation, task specific pretraining with public KITTI dataset and report segmentation accuracy together with computational efficiency on both training and generalization sets. Among the considered models, pretrained TransUNet achieved the strongest segmentation quality (IoU 0.929 ± 0.064, F1 0.962 ± 0.036), while U-Net offered a more deployable balance (IoU 0.921 ± 0.064, 80.31 ms/frame, 12.45 FPS, and the lowest environmental impact at 16.49 g $CO_2$e). We consider the trade-offs and deploy the system on multiple edge devices and validate it through outdoor prototype testing showing its practical effectiveness in real-world running scenarios. The system maintains stable lane detection and delivers timely haptic feedback at speeds of up to 10 km/h, demonstrating feasibility for autonomous vision based guidance in athletic running. To foster further development in this domain all code and data are publicly released (link: https://github.com/vrai-group/TrackAid-DT).

## 1. Introduction

Physical and sensory disabilities continue to limit the full participation of willing athletes in many sports (Wing et al., 2025). Although federations and events such as the Paralympic Games support athletes with disabilities, many activities rely on external guidance (Fig. 1), where athletes with visual impairment depend on sighted guides for lane discipline and safety. While effective, this practice inherently reduces autonomy and independence, especially during training sessions. Computer vision has long been explored for assistive technologies and navigation support (Leo et al., 2017). On running tracks, painted lane markings provide the primary visual cues for safe running but are not perceivable by athletes with visual impairment. Following this motivation, prior work demonstrated a chest-mounted, global-shutter monocular camera oriented slightly downward to stabilize egocentric capture during walking or jogging, with embedded processing to enable haptic guidance on standard athletic tracks in daylight (Mancini et al., 2018). This line of work established the feasibility of wearable, on-device lane perception for safe guidance.

Advancements in artificial intelligence, computer vision, and embedded hardware now enable devices that process visual input in

☆ This article is part of a Special issue entitled: 'CV for Sports' published in Computer Vision and Image Understanding.

* Corresponding author.
*E-mail address:* a.galdelli@univpm.it (A. Galdelli).

https://doi.org/10.1016/j.cviu.2026.104788
Received 7 October 2025; Received in revised form 17 April 2026; Accepted 18 April 2026
Available online 13 May 2026


**Fig. 1.** Athletes with visual impairment compete with the assistance of sighted guides (Davies, 2012).

real-time (Li et al., 2019; Liu et al., 2021; Wen et al., 2021). Efficient on-device semantic segmentation has progressed rapidly with compact backbones and deployment-oriented designs (Kim et al., 2024). Broader trends in edge computing also motivate low-latency inference (Afaq and Akram, 2025). These developments suggest that wearable capture setup can now benefit from modern deep networks that improve robustness while remaining deployable on embedded boards.

Studies have examined how embedded devices with navigational support and guidance for people with visual impairments can be useful, largely focused in activities of everyday or recreational contexts, for example, Han et al. (2023), detected blind lanes and obstacles, combining perception with potential-field control for safe sidewalk navigation. Extending this concept to sports environments, Kemeth et al. (2014), proposed a locating and guiding system for running on a track, while X. Liu et al. (2024), designed vision-based wearable steering assistance tailored to jogging on athletics tracks. These works demonstrate the feasibility of on-body vision systems, but their emphasis has largely been on mobility and recreational support. Such efforts highlight and prioritize coarse lane i.e., approximate estimation of lane position, path detection and prediction, obstacle avoidance, or user guidance for casual activities rather then from assistance in sports environment or its training.

Moreover, research in this area focuses mostly on system-level navigation heuristics (X. Liu et al., 2024; Mancini et al., 2018), leaving the interaction between segmentation quality, inference latency, and on-device feasibility not systematically studied. Therefore, the athletic perspective in terms of reliability has deteriorated, as visually impaired users have expressed dissatisfaction with the services provided by assistive technologies (Marques and Alves, 2021). The reason is that unlike general-purpose assistive devices (Han et al., 2023), systems for athletes must operate robustly at speed, under variable outdoor conditions, and with tight resource budgets on embedded hardware while remaining lightweight. However, direct and systematic attention to these requirements remains underexplored (X. Liu et al., 2024), in particular, two challenges stand out: (i) the limited availability of pixel-annotated datasets captured from the egocentric viewpoint required for on-body lane guidance, and (ii) the lack of systematic comparisons that balance segmentation accuracy, on-device latency, and energy consumption across heterogeneous embedded platforms within a single, implementable end to end system.

We target the same problem formulation, detecting track lane walkable areas from a wearable egocentric viewpoint on standard athletics tracks for safe guidance, and extend the perception stage with modern deep learning approaches. Specifically, we contribute: (i) a prototype vision-haptics system that integrates embedded inference with vibrotactile gloves to deliver lane-deviation feedback to visually impaired athletes; (ii) an egocentric dataset, TrackAid-DT with pixel-level track walkable area on the lane annotations collected outdoors on standard tracks under different conditions in athletic activities, following a chest-mounted capture setup and smartphone; (iii) a systematic comparison of multiple models under a unified preprocessing and training protocol with task specific pretraining on a public dataset, alongside eight-fold cross validation and a generalization test for robustness; (iv) conversion of model to TensorFlow Lite with post-training INT8 quantization, including full compilation on Edge TPU for real-time deployment; and (v) device-level latency, throughput, and energy/$CO_2$ measurements for all models and on Jetson Nano both natively and with an attached Coral Edge TPU accelerator. Finally, the system is evaluated on outdoor experiments of the proof of concept prototype demonstrating its practical effectiveness in real-world athletic running scenarios. Through these contributions, our work advances toward increased, participation, autonomy, safety and advanced studies for athletes.

## 2. Related work

This section discusses prior work in assistive vision and wearable guidance, lane detection approaches, efficient segmentation for edge deployment, and segmentation backbones, and concludes with the research gap in current literature.

### 2.1. Assistive vision and wearable guidance

Wearable assistive systems for people with visual impairment can be broadly categorized into three groups: mechatronic systems, infrastructure-based solutions, and vision-based approaches (Pieralisi et al., 2015; Mancini et al., 2018; X. Liu et al., 2024). Infrastructure-based approaches have been explored in structured environments, where electromagnetic tethering and guide systems support runner-lane alignment. For example, Pieralisi et al. (2015) designed a transmitter–receiver loop that encodes lateral deviation and provides vibrotactile feedback. While effective, such systems require instrumented environments and custom-built hardware (Pieralisi et al., 2015). More broadly, mechatronic and wearable sensing approaches emphasize on-body perception, low-latency processing, and interpretable feedback. Mancini et al. (2018) survey these systems in the context of mobility assistance, highlighting the importance of self-contained, real-time guidance solutions. More recently, camera-first wearable guidance has been proposed for non-competitive sports activities such as jogging scenarios, for instance, X. Liu et al. (2024) introduced a ground-perception and steering pipeline with a head mounted camera identifying issues arising from motion of visually impaired. This work focused on challenges including motion blur, rolling shutter artifacts, and rapid viewpoint

changes during casual jogging activities through the use of Multi-data for Athletics Track (MAT). This work also highlights several practical limitations for sustained athletic use. The head-mounted configuration amplifies high-frequency jitter and induces unstable visual input at running speed, complicating robust pixel-level perception. The reliance of the system on audio feedback, is not inclusive of athletes with listening impairment, as it relies exclusively on audio feedback for guidance. In addition, the proposed setup relies on sensing and processing pipeline based on the Jetson Orin NX. While this platform provides strong computational capabilities, the motivation of the experimental design is less oriented toward lightweight deployment, which is desirable in athletic training scenarios that may correspond to regular use (Wing et al., 2025). Consequently, the system is not lightweight, with an approximate total weight of around one kilogram. Furthermore, the approach primarily emphasizes line detection for guidance. For athletic use cases, an alternative focus could be placed on estimating walkable area in the lane. According to World Athletics Competition Rules (World Athletics, 2024), athletes are required to remain within their assigned lane throughout the race (Rule 17, Track Events), and any encroachment that causes interference/results in a competitive advantage constitutes a violation (Rule 18, Obstruction), which may lead to disqualification depending on race conditions. These rules are also adopted within World Para Athletics regulations, with class-specific adaptations (World Para Athletics, 2024). Collectively, these works motivate athlete-centric, wearable viewpoints for guidance, but they either depend on external infrastructure (Pieralisi et al., 2015), prioritize general mobility assistance with limited learning-based perception (Mancini et al., 2018), or present early-stage egocentric designs that face robustness, stability, and deployment challenges under realistic athletic dynamics (X. Liu et al., 2024). These limitations highlight the need for wearable vision systems that balance the motivation, accuracy, viewpoint stability, and on-device efficiency specifically for running on standard athletic tracks.

Moreover, sports-specific lane perception remains sparse. Panoptic driving systems, which perform full-scene pixel-level understanding by combining semantic and instance segmentation, include a lane head demonstrate embedded real-time performance and multi-task synergy, yet they are tuned to urban scenes and wider lane geometries rather than narrow track markings (Wu et al., 2022). In the case of assistive guidance in sports environment, segmenting the entire area accessible for the athlete with visual impairment provides more comprehensive spatial supervision and improves generalization in such conditions. The problem formulation is consistent with prior work on free-space and drivable-area detection (Yu et al., 2020), that focused on areas confined within markings, and additionally looking at the approach of Patra et al. (2018) demonstrating that reliable navigation can be achieved without relying on lane markings.

### 2.2. Efficient segmentation and edge deployment

Real-time semantic and instance segmentation have advanced significantly through architectural and compression innovations. BiSeNet decouples spatial and context paths, improving boundary quality useful for thin markings at 100+ FPS on commodity GPUs (Yu et al., 2018). Mobile-optimized backbones such as MobileNetV3 and GhostNet reduce Multiply Accumulate (MAC) operations, which is a common measure of computational cost that counts the number of weight–input multiplications and additions a model performs. Fewer MACs translate into lower latency and energy use, making such models attractive for resource-constrained devices (Han et al., 2020; Howard et al., 2019). Quantization techniques enabling integer-only inference further support efficient deployment on embedded processors, including ARM-based systems and digital signal processors (Jacob et al., 2018), while pruning, quantization, and coding strategies (“deep compression”) reduce memory footprint and energy requirements (Han et al., 2016). End-to-end multi-task pipelines designed for automotive edge devices provide instructive references, for instance, YOLOP runs object/drivable-area/lane heads on shared encoders, underscoring careful capacity allocation across tasks (Wu et al., 2022). Still, athlete-worn cameras exacerbate challenges of motion blur, frequent pose changes, specular track textures, and highly curved near-field lanes consequently stressing models validated under smoother vehicle dynamics. The literature has received relatively limited attention on accuracy/latency/energy trade spaces under egocentric motion or for thin, high-curvature markings with strong roll/pitch. On the other hand, U-Net and its derivatives have demonstrated strong performance in scenarios where precise pixel level boundary delineation is critical. In parallel, the recent dominance of Transformer and Mamba based architectures for modeling long range dependencies suggests that these paradigms are particularly effective in complex, high curvature geometries (Yao et al., 2024). However, despite their significant performance gains, the practical trade-offs of these architectures remain insufficiently explored for egocentric sports guidance, where accuracy requirements are comparable to those in healthcare (Kim et al., 2024), yet efficiency trade-offs (Zhang et al., 2024) or deployment constraints and edge level considerations introduce distinct challenges.

### 2.3. Segmentation backbones relevant to lane marking

Backbones that preserve fine structures while capturing long-range context are especially relevant for lane boundaries. UNet++ redesigns skip connections and leverages deep supervision to improve multi-scale fusion advantages for sub-pixel markings and crisp boundaries (Zhou et al., 2020). Transformer-CNN hybrids such as TransUNet inject global self-attention into U-shaped decoders, improving context aggregation for thin structures without sacrificing localization (Chen et al., 2021). Recently, visual state-space models, such as VMamba, offer linear-time selective scanning over 2D features and competitive segmentation quality, promising long-range dependencies with favorable memory or latency scaling (Y. Liu et al., 2024). For lane tasks, these encoders-decoders can be paired with structure-aware heads (row-wise, anchor-based, polynomial) to balance detail fidelity and global continuity. Knowledge distillation tailored for lanes (Self Attention Distillation) compresses student networks while retaining topology cues, mitigating accuracy drops in tiny models (Hou et al., 2019). In practice, combining a fine-structure-friendly decoder, an efficient encoder and deployment-aware quantization (INT8) yields strong edge baselines, but behavior under egocentric athletic dynamics remains unreported.

### 2.4. Research gap

Across assistive wearable, the literature affirms the value of on-body sensing and low-latency feedback but seldom targets athletic track lanes; infrastructure-heavy solutions are impractical outside controlled events (Mancini et al., 2018; Pieralisi et al., 2015). Automotive lane detection offers rich algorithmic toolkits such as SCNN/RESA for continuity, UltraFast/LaneATT for efficient decoding, PolyLaneNet for compact parametrics, and LaneNet for instance separation, that are, validated on TuSimple, CULane, LLAMAS, and BDD100K (Behrendt and Soussan, 2019; Neven et al., 2018; Pan et al., 2018; Qin et al., 2020; Tabelini et al., 2021b,a; Yu et al., 2020; Zheng et al., 2021). On the other hand, these datasets assume vehicle viewpoints and lane geometries unlike athletic tracks. Efficiency work (BiSeNet, lightweight backbones, integer quantization, compression) is compelling for wearables but typically benchmarked on automotive SoCs or desktops rather than head/torso-mounted cameras with rapid pose changes (Han et al., 2020, 2016; Howard et al., 2019; Jacob et al., 2018; Yu et al., 2018). Modern backbones (UNet++, TransUNet, VMamba) promise improved thin-structure fidelity and long-range context at competitive cost, but their accuracy-latency-energy trade-offs for egocentric running videos

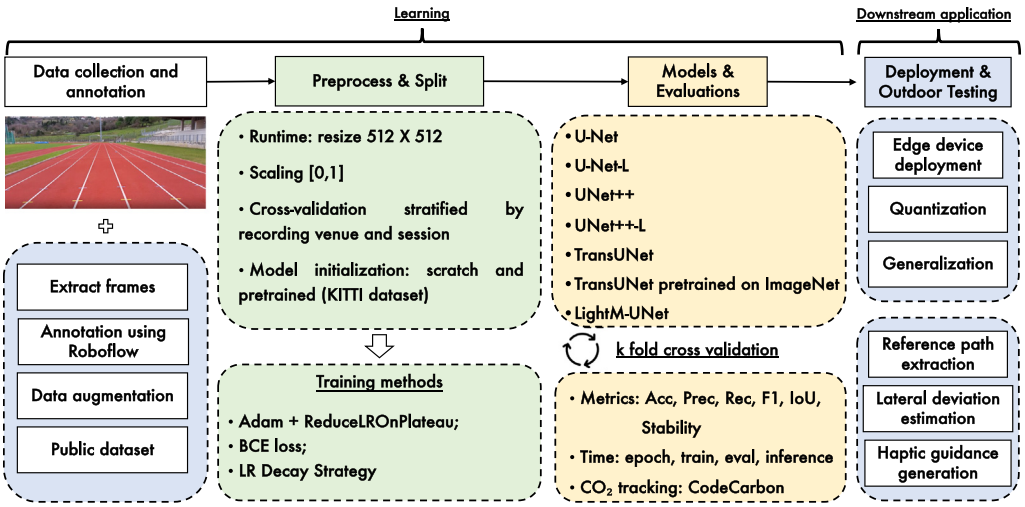



**Fig. 2.** Proposed pipeline overview.

are uncharacterized (Chen et al., 2021; Y. Liu et al., 2024; Zhou et al., 2020), that makes advanced investigations challenging.

To the best of our knowledge, current datasets and methods do not systematically address the perception of walkable track lane areas from chest-mounted wearable cameras during athletic running. Existing approaches for assistive navigation primarily focus on walking or jogging scenarios (X. Liu et al., 2024), or rely on viewpoints and sensing configurations that differ substantially from the egocentric (Wing et al., 2025), on-body perspective induced by high-speed athletic motion. As a result, the impact of strong pose variations, near-field perspective, rapid heading changes, and highly curved lane geometry on perception performance remains largely unexplored. Moreover, prior chest-mounted systems employ classical or hand-crafted modeling pipelines (Mancini et al., 2018), providing limited insight into the behavior of modern deep learning-based segmentation architectures under these challenging conditions. In particular, there is a lack of systematic evaluation of structure-aware convolutional, transformer-based, and state-space models when trained and tested on data captured during athletic motion, as well as insufficient analysis of their generalization capabilities under data-limited regimes. Equally important, existing studies rarely examine the practical constraints imposed by wearable deployment. The trade-offs between segmentation accuracy, inference latency, and energy consumption on embedded edge devices remain under-characterized, despite being critical for continuous, on-body operation. This gap is further amplified by the absence of end-to-end evaluations that integrate egocentric perception with real-time haptic feedback and validate the feasibility of such systems in realistic athletic settings. The present work is therefore motivated by the need for a comprehensive investigation that jointly considers data, perception models, deployment constraints, and system-level validation within a unified experimental framework.

## 3. Materials and methods

This section details our proposed pipeline (Fig. 2) and focuses on procedures of data acquisition, preprocessing, model architectures, training, evaluation metrics and deployment strategy used in our study.

### 3.1. Wearable egocentric vision system

The recording configuration was designed for reliable and stable acquisition, following the guidelines of Mancini et al. (2018). Specifically, as showcased in Fig. 3, a monocular BlueFox MLC200wC global-shutter camera was chest-mounted and tilted downward by ∼10° to stabilize egocentric capture and reduce motion jitter relative to head-mounted setups. The camera streamed frames to NVIDIA® Jetson Nano edge device board, where the vision pipeline and steering decisions are executed. Commands were transmitted wirelessly to gloves via Bluetooth Low Energy. Each glove integrated a custom PCB with a Bluegiga BLE112 module (microcontroller and BLE radio with chip antenna), a 950 mAh Li-Po battery, and two Parallax 12-06B002F eccentric rotating mass vibration motors placed on the thumb and index finger. The BLE112 generated pulse width modulation signals i.e., electrical pulses with duration encoding the desired vibration strength which is filtered and amplified to drive the motors. Firmware services were implemented in BGScript, including the Generic Access Profile (GAP), Battery Service, and a Custom Service, which enabled the chest unit to wirelessly trigger independent left or right vibrations with low latency. The BLE112 generated pulse-width modulation signals, which were filtered and amplified to drive the motors, allowing precise control of vibration intensity. The complete wearable setup had an overall weight of approximately 750 g, dominated by the chest-mounted processing unit and battery, while the exact weight varied slightly depending on the battery configuration. The battery configuration that we used is of 950 mAh Li-Po battery, sufficient for several hours of use. This way, the system delivered directional cues that guided the athlete's steering by selectively activating the motors on the left or right glove. The glove design is kept unobtrusive, ensuring that athletes could still maintain a natural running posture while receiving continuous tactile feedback. The sensor delivers 752 × 480 frames up to 90 Hz; in prior work the downstream pipeline was operated around 20 Hz to match user reaction times during jogging.

### 3.2. Data acquisition

Data were collected with three athletes on standard outdoor tracks (in Marche province of central Italy at Ancona and Fermo) with painted lane boundaries under diverse illumination, direct sunlight with strong shadows, cloudy or evening light, and nighttime under artificial lighting to reflect realistic deployment. During trials, the athletes ran at standard training speeds (appx. up to 2–3 m/s). Low-light conditions were specifically included to represent challenging scenarios for visually impaired athletes, such as evening runs without direct illumination and environments lit only by nearby lamps. Beyond illumination, the sessions also varied in viewpoint and motion. Close-up and far-view perspectives naturally arose from the runner's lateral offset within the lane, while sprinting sessions (faster than 3 m/s) introduced stronger egomotion compared to steady jogging. Furthermore, the dataset was augmented with a supplementary set of videos captured via handheld smartphone cameras, providing a strategic contrast to the primary

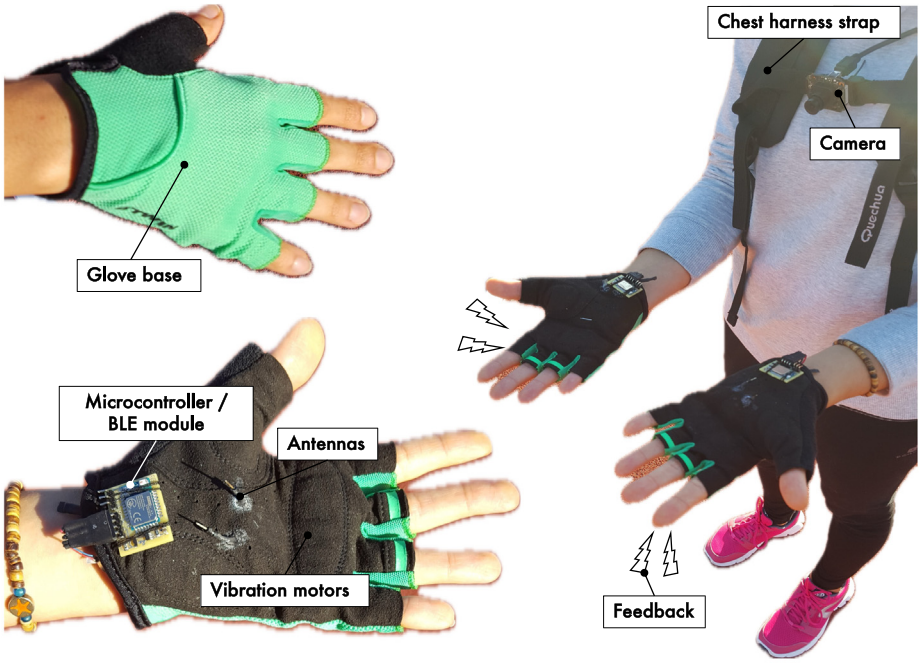



**Fig. 3.** Wearable egocentric vision system.

chest-mounted configuration. These sequences, which include both portrait and landscape orientations, were deliberately added to expose the models to heterogeneous acquisition conditions and better generalization, reflecting the variability encountered in real-world deployment scenarios, as suggested by expert evaluations (Geiger et al., 2012). Together, these variations ensured the dataset captured the range of practical difficulties that visually impaired athletes would face in real training settings. Representative frames in the prior studies (X. Liu et al., 2024; Mancini et al., 2018; Liu et al., 2021) show that bright sun and low visual signal to noise are the hardest conditions; our dataset includes additional similar cases to stress robustness on the models.

### 3.3. Dataset

Compared to previous study (Mancini et al., 2018), our collected dataset extends by incorporating videos from three additional athletes and a broader range of recording sessions. While the earlier dataset, acquired in two different venues (Ancona and Fermo), was sufficient for evaluating classical image-processing pipelines, but such limited scale is restrictive for training modern deep architectures. By including more athletes and more varied conditions (day, night, low light, sprint pace, and different viewing geometries as illustrated in Fig. 4), the present dataset offers greater diversity and more realistic coverage of deployment scenarios. From the recorded videos, frames were uniformly sampled at a rate of one frame every three seconds, yielding 747 images that were subsequently annotated with pixel-level lane masks using Roboflow[1]. Each image is paired with a binary segmentation mask of the central lane interior. To increase robustness, we applied data augmentations (only during the training of our models) including random rotations, horizontal flips, brightness and contrast shifts, and Gaussian noise injection. More advanced approaches, such as controllable inpainting with generative models, have been proposed to synthetically enrich domain-specific datasets and overcome annotation scarcity (Biancucci et al., 2025). Although our study does not employ generative augmentation, such methods offer additional promising directions for expanding training diversity. The carried augmentations simulate variations in runner posture, viewpoint, lighting, and sensor noise, thereby improving model generalization beyond the limited base set. The dataset reflects the qualitative variability of real track scenes.

*Annotation protocol.* Each frame was labeled with a binary mask of the central lane (lane = 1, background = 0) using polygon-based annotation (as shown in Fig. 5). Annotators were instructed to include the drivable interior of the current lane and to exclude adjacent lanes, dashed boundary paint outside the lane, and off-track regions. Ambiguous frames were flagged for second-pass review. We applied a two-stage quality process: (i) intra-annotator self-checks for boundary consistency and closure of polygons; (ii) reviewer pass focusing on the handling of curves, faint markings, and transitions between straight and curved segments. The final dataset comprises a compact set of labeled images in the order of hundreds of frames extracted from multiple capture sessions. We ensured diversity by sampling evenly across venues, lighting conditions, and track geometries. Frames were downsampled from video to limit near-duplicates; nonetheless, sequential correlation is expected within a session due to egocentric capture. To prevent data leakage, the dataset was partitioned by recording sessions, ensuring that frames from the same session never appear in different folds. The prediction target is a per-pixel mask of the central lane area rather than thin boundary lines. This choice simplifies downstream control (e.g., centerline estimation and lateral deviation) and is robust to partial occlusions of the paint. To prevent train-mismatch, all preprocessing steps applied during training are identical at evaluation time.

### 3.4. Egocentric guidance and haptic actuation

The processing unit in the proposed system corresponds to the chest-mounted NVIDIA® Jetson Nano edge device. It hosts the complete software stack for image acquisition, segmentation inference, guidance computation, and wireless communication. The processing unit implements the BLE communication stack and a custom BLE service used to transmit actuation commands to the gloves. All sensing, computation, and guidance logic are centralized on the processing unit, while the gloves contain only the BLE112 microcontroller, vibration motors, and supporting power electronics. For each acquired frame, the processing

[1] https://www.roboflow.com, Accessed: Accessed: 15 April 2026

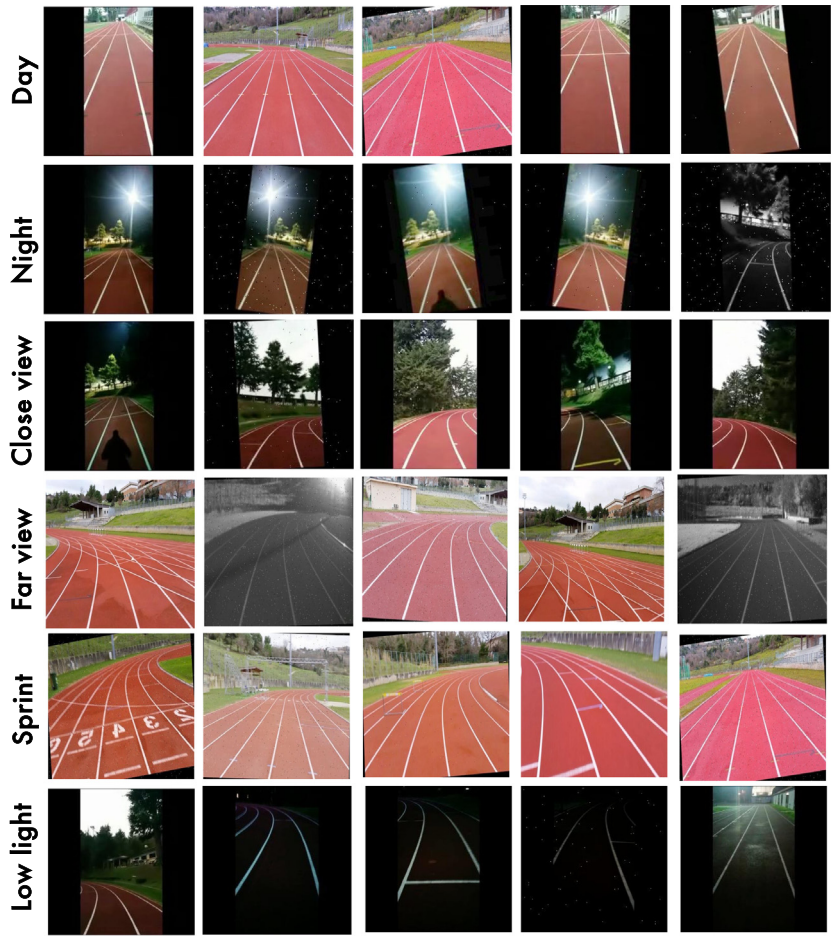



**Fig. 4.** Data set along with the augmentation and challenging conditions.

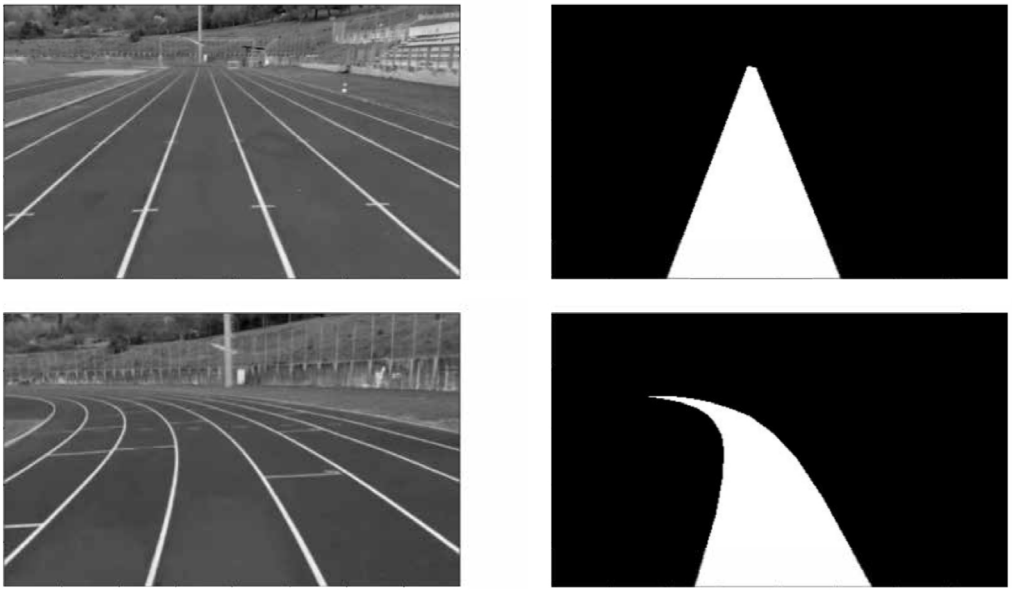


**Fig. 5.** Examples of dataset labels.

unit takes the segmented image (obtained from quantized segmentation model) to estimate a geometric reference associated with the runner's desired path. In lane-tracking mode, the reference is defined as the centerline of the segmented walkable lane. When multiple parallel lane markings are visible, the system maintains temporal consistency by continuing to track the initially selected lane as long as both boundaries remain detectable. If one boundary disappears, the reference is redefined using the most stable available segmented region. In this

formulation, following the guidelines of Mancini et al. (2018), we consider the path reference as the centerline of the detected lane. The image plane is defined by width $w$ and height $h$. The lateral deviation estimation is evaluated at a fixed vertical coordinate near the bottom of the image, corresponding to the closest visible ground region to the athlete. The lateral error is defined as

$$e_t = x_r(h) - \frac{w}{2}, \tag{1}$$

where $x_r(h)$ denotes the horizontal image coordinate of the reference at image row $y = h$. This quantity measures the instantaneous lateral offset of the runner with respect to the desired path in the egocentric image plane. To capture short-term directional changes, a discrete orientation proxy is derived from the temporal evolution of the lateral error. Let $m$ denote the frame index. The orientation signal is defined as

$$e_\theta(m) = e_t(m) - e_t(m-1), \tag{2}$$

and smoothed using an exponential moving average, implementing temporal smoothing across consecutive frames. The exponential moving average acts as a temporal smoothing mechanism, reduces the frame-to-frame noise and stabilizes the guidance signal during running, and is formulated as,

$$\bar{e}_\theta(m) = (1-\alpha)\bar{e}_\theta(m-1) + \alpha e_\theta(m), \tag{3}$$

where $\alpha$ controls responsiveness relative to human reaction times during running. A local curvature proxy is derived using a look-ahead formulation. Let $l_d$ denote a fixed look-ahead distance along the reference path and $y_g$ the lateral displacement of the reference at that distance. The instantaneous curvature is approximated as

$$\kappa = \frac{2y_g}{l_d^2}. \tag{4}$$

The lateral and orientation deviation signals are interpreted within a control framework that aims to drive the runner back toward the desired path. The controller models the runner as a differential-drive agent and generates corrective actions aimed at reducing both lateral displacement and heading misalignment with respect to the reference path through a haptic command mapping through intensity of vibration. Corrective actions are derived by jointly considering the magnitude of the lateral error and the temporal evolution of the orientation signal, allowing the system to determine the direction and strength of the feedback to be delivered. The controller output is translated into vibrotactile commands, where the active glove encodes the direction of correction and the vibration characteristics encode the required correction intensity. This formulation enables continuous guidance during walking or running by providing incremental directional cues that drive the user toward the desired path. Directional vibrotactile guidance is generated based on the sign and magnitude of the deviation signals. Positive lateral deviation activates the left glove to indicate a rightward correction, while negative deviation activates the right glove to indicate a leftward correction. The *clip()* function ensures that the vibration intensity $I$, computed from the lateral error $e_t$ and the filtered orientation error $\bar{e}_\theta$, remains within the minimum and maximum admissible intensity levels of the haptic device. The *clip()* operator corresponds to a saturation nonlinearity commonly adopted in control and haptic systems to constrain the control signal within the physical and perceptual limits of the actuator (Khalil, 2002; Åström and Murray, 2008). The vibration intensity $I$ is computed as

$$I = \text{clip}\left(k_t|e_t| + k_\theta|\bar{e}_\theta|, I_{\min}, I_{\max}\right), \tag{5}$$

where $k_t$ and $k_\theta$ weight the lateral and directional components, and $I_{\min}$ and $I_{\max}$ correspond to the hardware limits of the vibration motors. To avoid oscillatory feedback when the runner is close to the desired path center, actuation, is limited to a minimum intensity, when $|e_t| < \varepsilon$, where $\varepsilon$ defines an acceptable central tolerance region. Additional vibration patterns are reserved for discrete system states, such as loss of visual reference or temporary guidance suspension, and can be extended to encode speed increase or decrease commands, while further configuration modes and vibrotactile mappings are described in a prior work (Mancini et al., 2018). Actuation commands are transmitted over Bluetooth Low Energy using the custom service, and pulse-width modulation encodes vibration intensity at the glove level. The achievable running speed is constrained by the end-to-end latency of image acquisition, onboard processing, and wireless actuation, ensuring that guidance cues remain interpretable during continuous motion.

### 3.5. Preprocessing

The preprocessing stage is designed to ensure consistency across training, validation, and testing while remaining lightweight enough for deployment on embedded hardware. In classical approaches, depending on the track surface characteristics, frames are processed either in grayscale or in HSV color space (Mancini et al., 2018). In contrast, deep learning models improve on this aspect and operate directly on the input representation and learn relevant features automatically, reducing the need for manual selection of color spaces. The raw data for our case consist of RGB images, and all input frames are standardized through a fixed preprocessing pipeline applied uniformly to ensure consistency across models and enable fair comparison. Every frame is resized to $512 \times 512$ pixels using bilinear interpolation for images and nearest-neighbor interpolation for masks, which prevents class mixing along lane boundaries. This uniform resizing ensures that all models receive inputs of identical dimensions, simplifying training across architectures and enabling fair comparisons of inference latency. Pixel intensities are globally scaled to the interval $[0, 1]$. Unlike normalization schemes that compute dataset- or image-specific means and variances, we deliberately adopt this fixed scaling to preserve stable activation ranges across datasets. The same pipeline is invoked at test time and for on-device deployment, ensuring that the numerical conditions under which latency and accuracy are measured faithfully reflect the conditions of embedded inference. Finally, model predictions are thresholded at 0.5 to obtain binary segmentation masks, and no morphological post-processing is performed.

### 3.6. Model architectures

The choice of network architecture plays a central role in the segmentation of lane markings, which are thin, elongated, and subject to background clutter or motion blur. To provide a fair comparison across design philosophies, we implemented a family of encoder–decoder models in TensorFlow/Keras and trained them at a fixed input resolution of $512 \times 512$ pixels. This resolution balances spatial detail and computational feasibility, preserving fine-grained lane structures within practical memory and runtime limits, while remaining consistent with the Transformer-based TransUNet design, which operates on square feature partitions for this input size. To maintain fairness across comparisons, the same resolution is used for all models, ensuring that differences in accuracy and latency arise from architectural choices rather than mismatched input scaling.

*U-Net and U-Net-L.* U-Net serves as a baseline architecture due to its proven ability to capture thin structures through its symmetric encoder–decoder pathway and dense skip connections (Ronneberger et al., 2015). These skips directly transfer high-frequency details from early layers to later reconstruction stages, which is especially valuable when detecting narrow lane boundaries. The lightweight variant (U-Net-L) reduces the encoder–decoder depth by one stage, thereby cutting parameter count and latency. This makes U-Net-L particularly suitable for scenarios where inference must run on constrained devices, while still preserving the structural inductive bias of the original design.

*UNet++ and UNet++-L.* UNet++ extends this principle by introducing nested skip pathways (Zhou et al., 2018) that include intermediate convolutional blocks. These dense pathways narrow the semantic gap between encoder and decoder features, yielding more effective multi-scale fusion. For thin and fragmented targets such as lane markings, this design helps mitigate the loss of detail across scale transitions. The light variant (UNet++-L) follows the same philosophy as U-Net-L, applying depth reduction to contain computational cost while retaining the nested skip structure that distinguishes UNet++. It does so by removing one encoder–decoder stage while retaining the nested skip structure. This depth reduction lowers parameter count and latency while preserving the key design principles of UNet++.

*TransUNet (scratch) and TransUNet (pretrained on ImageNet).* While convolutional encoders excel at local detail, they can struggle to capture long-range dependencies, particularly when contextual cues such as road geometry are critical. TransUNet addresses this by combining a CNN backbone with a Transformer encoder, followed by a U-Net-style decoder (Chen et al., 2024). The Transformer module models global spatial relationships, complementing the local precision of the CNN. We examine two variants: one trained from scratch with randomly initialized weights, and one initialized from ImageNet-pretrained weights. ImageNet is a large-scale dataset of annotated natural images spanning thousands of object categories, including animals, vehicles, tools, and common objects (Deng et al., 2009). This initialization provides generic visual features and provides a low- and mid-level visual features, which can improve generalization when fine-tuned on the target segmentation task.

*LightM-UNet (Mamba-based).* LightM-UNet represents a more recent design direction that replaces self-attention with state-space models, specifically Mamba blocks, which capture long-range dependencies with near-linear complexity (Liao et al., 2024). This reduces computational overhead compared to Transformer-based attention while retaining the ability to model spatial correlations beyond the local receptive field. Our implementation simplifies the state space model module to reduce memory usage during training, for experimentation on embedded hardware, while preserving skip connections to safeguard spatial fidelity.

We investigate the effect of task-specific pretraining using the KITTI dataset (Fritsch et al., 2013), which provides annotated road and lane structures. The dataset contains road scenes with relatively low traffic density, where lane markings are visible, providing clean structural supervision and domain priors for lane geometry relevant to our application of interest in lane detection. For a systematic comparison, we employ a task-specific pretraining stage with KITTI dataset and subsequently fine-tune on TrackAid-DT. This pretraining phase is separate to the generic backbone initialization used in TransUNet architecture, where the encoder is initialized with ImageNet weights. KITTI based task specific pretraining introduces structural priors related to road geometry and lane boundaries. This distinction allows us to analyze the effects of generic and task aligned feature initialization within a unified experimental framework. To this end, all architectures are evaluated under two training settings: training from scratch on TrackAid-DT, and with an intermediate pretraining stage on KITTI followed by fine-tuning on TrackAid-DT. This design, as investigated also in prior study in assistive technology implementation (X. Liu et al., 2024), allows us to systematically assess the impact of domain-aligned pretraining on segmentation performance while ensuring a consistent and fair comparison across models. All architectures output a probability map $P \in [0,1]^{H\times W}$, where each entry reflects the estimated likelihood of a pixel belonging to the lane class. For evaluation, these maps are thresholded at 0.5 to obtain binary segmentation masks. Dropout and batch normalization layers are included selectively to stabilize training without complicating inference. Importantly, all models share the same preprocessing pipeline, loss function, optimizer, and learning-rate scheduling strategy, ensuring that differences in performance can be attributed to architectural choices rather than training protocol.

**Table 1**
Final training schedules per architecture. LR-decay indicates the multiplicative factor used by ReduceLROnPlateau.

| Architecture | Epochs | LR | MinLR | Patience | LR-decay | Batch |
|---|---|---|---|---|---|---|
| U–Net | 100 | $5\times10^{-5}$ | $1\times10^{-6}$ | 3 | 0.5 | 8 |
| U–Net–L | 100 | $1\times10^{-4}$ | $1\times10^{-6}$ | 4 | 0.5 | 8 |
| UNet++ | 100 | $2.5\times10^{-5}$ | $1\times10^{-7}$ | 3 | 0.5 | 8 |
| UNet++–L | 100 | $5\times10^{-5}$ | $5\times10^{-7}$ | 3 | 0.4 | 8 |
| TransUNet | 100 | $1\times10^{-4}$ | $5\times10^{-7}$ | 3 | 0.5 | 8 |
| PretrainedTransUNet[a] | 100 | $1\times10^{-4}$ | $5\times10^{-7}$ | 3 | 0.5 | 8 |
| LightM–UNet | 100 | $1\times10^{-5}$ | $1\times10^{-7}$ | 3 | 0.5 | 8 |

[a] PretrainedTransUNet denotes the model initialized with ImageNet-pretrained weights.

### 3.7. Training settings and hyperparameter tuning

The experimental setup was conducted on a workstation equipped with an NVIDIA® Quadro RTX 6000 GPU with 48 GB of RAM. All models were implemented using Python v3.10. To support reproducibility and further research, the full implementation has been made publicly available as an open-source repository on GitHub[2]. All architectures were trained under a shared protocol that standardized preprocessing, optimizer choice, and callback configuration, with model-specific schedules summarized in Table 1. Each model was trained for up to 100 epochs with early stopping monitored on the validation loss and a patience of 3–4 epochs. Learning rates were decayed on plateau using a multiplicative factor of 0.5 for all models except UNet++-L, which used 0.4. Initial learning rates varied from $1\times10^{-5}$ in LightM-UNet to $1\times10^{-4}$ in U-Net-L, TransUNet, and its ImageNet pretrained variant, while minimum learning rates between $1\times10^{-7}$ and $1\times10^{-6}$ were enforced by the scheduler. Each network is trained at a fixed input resolution under a similar 8-fold cross-validation scheme. The dataset was partitioned into eight non-overlapping folds of approximately equal size, each corresponding to a distinct video sequence. At each iteration, one fold was held out for validation, while the remaining folds were used for training, following the same early stopping and learning rate scheduling criteria described above. This procedure was repeated until each fold had served once as the validation set. Final performance metrics are reported as the mean and standard deviation across all folds, providing a robust estimate of model performance in data-limited settings and reducing sensitivity to a specific data split. All the models were trained using a batch size of 8, optimized with the Adam optimizer, and supervised with binary cross-entropy loss, ensuring a consistent training protocol across architectures. During evaluation, sigmoid outputs were binarized at a fixed threshold of 0.5, ensuring that reported metrics are directly comparable across models.

Given the limited size of the TrackAid-DT dataset, we additionally evaluate all our models through a task specific pretraining with public dataset. The evaluation was conducted using similar 8-fold video-level cross-validation strategy that was used for training from scratch. For transfer learning experiments, models were first pretrained on the KITTI Road and Lane Detection dataset (Fritsch et al., 2013), which comprises approximately 1200 images after data augmentation, and subsequently fine-tuned within each cross-validation fold using only TrackAid-DT samples. Models trained from scratch followed the same cross-validation protocol, enabling a direct and fair comparison between training strategies under identical evaluation conditions. The use of KITTI (Fritsch et al., 2013) for pretraining follows established practice in computer vision (X. Liu et al., 2024; Alfano et al., 2024), allowing models to learn task specific features prior to adaptation, which for our case, through KITTI offers a good balance of multiple scene types for lane segmentation, complementing our TrackAid-DT. Finally, to assess out-of-distribution generalization test, all trained models were evaluated on a fully independent external test set that was not involved in either the cross-validation or training process, providing an additional measure of robustness beyond in-distribution performance.

[2] https://github.com/vrai-group/TrackAid-DT

### 3.8. Evaluation protocol and metrics

The input data follow the preprocessing pipeline in Section 3.5; to ensure reproducibility, we fix the random seed in dataloading and metrics reported as the mean and standard deviation across folds. The primary accuracy measure is pixel accuracy on the binary task (lane versus background). Let $TP$, $TN$, $FP$, and $FN$ denote the number of true-positive, true-negative, false-positive, and false-negative pixels. Pixel accuracy is

$$\text{Acc} = \frac{TP+TN}{TP+TN+FP+FN}.$$

For completeness, we also compute precision, recall, F1-score, and Intersection-over-Union (IoU) for the lane class:

$$\text{Prec} = \frac{TP}{TP+FP}, \quad \text{Rec} = \frac{TP}{TP+FN}, \quad \text{F1} = \frac{2\,\text{Prec}\cdot\text{Rec}}{\text{Prec}+\text{Rec}},$$
$$\text{IoU} = \frac{TP}{TP+FP+FN}.$$

The downstream task of lane guidance depends on both the accuracy and the temporal stability of the detected walkable region. To account for this, we introduce a complementary metric, termed Stability. This metric assesses the relative performance ordering of models across folds and the consistency of their segmentation behavior under cross-validation. The metric reports the standard deviation of IoU across folds, which reflects the consistency of model performance under different data splits.

All the reported metrics are for the foreground (lane) class, as it directly corresponds to the target of the segmentation task. When reported as mean metrics, we average the per-class values across lane (foreground) and background pixels separately. The additional results for background class metrics are then provided in the Appendix for completeness. To relate segmentation quality to deployability, we pair accuracy with median per-frame latency at batch size 1 on each target device (GPU, Jetson Nano CPU, and Coral Edge TPU). Latency is measured from input tensor ingestion to probability map output after a short warm-up. All measurements are taken with the identical preprocessing path used during training to align expectations between offline evaluation and on-device runtime. We used the CodeCarbon framework (Lacoste et al., 2019), to quantify energy and environmental cost, which estimates emissions based on power draw (CPU, GPU, and memory utilization) measured at runtime, combined with the carbon intensity of the regional electricity grid. For each experiment, we log both training and evaluation phases, and report the sum of their emissions as the total footprint of a model. Emissions are expressed in grams of $CO_2$ equivalent (g $CO_2$e), a standardized unit representing the greenhouse impact of the consumed energy. Physically, this value corresponds to the estimated mass of $CO_2$ released into the atmosphere due to electricity consumption during computation, and provides a reproducible measure to compare the environmental cost of different architectures and deployment strategies.

## 4. Results

We evaluate seven segmentation architectures on the dataset: U-Net, U-Net-L, UNet++, UNet++-L, TransUNet (scratch and pretrained on ImageNet), and LightM-UNet. Results are reported for models trained from scratch as well as for models initialized with KITTI pretraining, with all experiments conducted under the same 8-fold cross-validation protocol.

### 4.1. Segmentation accuracy

The results of the 8-fold cross-validation in terms of IoU, F1-score, Recall, Precision, Accuracy, and Stability and associated standard deviation of models trained on scratch and pretrained with KITTI dataset are presented in Table 2. In the case of training from scratch the TransUNet (pretrained on ImageNet) achieves the highest mean IoU of $0.929 \pm 0.064$. TransUNet follows with a mean IoU of $0.865 \pm 0.120$. UNet++ and U-Net report comparable mean IoUs of $0.859 \pm 0.113$ and $0.854 \pm 0.102$, respectively. UNet++-L and U-Net-L achieve mean IoUs of $0.763 \pm 0.165$ and $0.738 \pm 0.178$. LightMUNet records a mean IoU of $0.697 \pm 0.182$. The results with a task specific pretraining on KITTI dataset shows that U-Net achieves the highest mean IoU of $0.921 \pm 0.064$, followed by UNet++ with $0.917 \pm 0.067$. TransUNet (pretrained on ImageNet) records a mean IoU of $0.899 \pm 0.073$. U-Net-L and UNet++-L obtain mean IoUs of $0.884 \pm 0.089$ and $0.815 \pm 0.121$, respectively. TransUNet achieves a mean IoU of $0.796 \pm 0.136$. LightMUNet reports the lowest mean IoU of $0.578 \pm 0.244$ and the highest IoU variability. Accuracy values across all models remain above 0.918.

The experimental results highlight a heterogeneous impact of KITTI-based pretraining across the evaluated segmentation architectures. While convolutional encoder–decoder models belonging to the U-Net family consistently benefit from pretraining, transformer-based hybrid models exhibit a different behavior, achieving comparable or superior performance when trained from scratch on the target dataset. In particular, U-Net and UNet++ show the highest performance under the fine-tuning regime, both reaching IoU and F1 scores above 0.92 and 0.95 respectively. These improvements are accompanied by a relatively low standard deviation across cross-validation folds, indicating higher accuracy and greater robustness to variations in the input data. This behavior is attributed to the hierarchical convolutional feature extraction mechanism of UNet-like architectures, which relies on low-level visual primitives such as edges, contours, and texture patterns. Such features are transferable across domains, making larger datasets like KITTI effective sources for task specific pretraining even with difference in the downstream task context. Conversely, transformer-based architectures such as TransUNet (scratch and pretrained) and LightMUNet demonstrate improved performance when trained from scratch on the target dataset. In these cases, pretraining on KITTI does not consistently lead to performance gains and, in some configurations, results in lower IoU and F1 scores compared to scratch training. This behavior suggests the presence of negative transfer effects, due to the strong reliance of attention-based models on global contextual relationships. Since KITTI contains urban driving scenes that are characterized by structured layouts and rigid objects, the global attention patterns learned during pretraining do not align well with the visual characteristics of lane running area detection, which is dominated by curvature in shapes, occlusions on ground (for instance hurdles), and other texture-driven cues for the athlete. The analysis of Stability metric across cross-validation folds gives additional insight into model behavior. Architectures benefiting from pretraining (U-Net and UNet++) exhibit lower variability in performance, indicating consistent generalization across different sequences. This is suggestive of the fact that pretraining contributes to higher mean accuracy and more consistent representations, reducing sensitivity to fold-specific characteristics. In contrast, models trained from scratch, especially transformer-based ones, tend to show higher standard deviation, reflecting a stronger dependence on the specific data distribution observed during training. Extended per-class metrics (Table A.7) confirm that performance gains are concentrated on the foreground (lane) class while background segmentation remains uniformly strong, with most variations arising due to model complexity and challenging lighting conditions.

Visual examples in Fig. 6 reflect the quantitative performance trends reported in Table 2. Under favorable conditions such as daytime recordings and close-view sequences, most architectures generate clean lane masks with minimal artifacts, consistent with the high overall accuracy observed across models. In more challenging scenarios, including night runs, rapid motion, and low-light environments, performance differences become more evident. Models with higher IoU and lower variability such as the pretrained TransUNet, preserve lane segmentation area continuity and produce more consistent predictions under

**Table 2**
Model performance under cross-validation training from scratch and KITTI pretraining.

| Model | IoU ($\mu \pm \sigma$) ↑ | F1 ($\mu \pm \sigma$) ↑ | Recall ($\mu \pm \sigma$) ↑ | Precision ($\mu \pm \sigma$) ↑ | Accuracy ($\mu \pm \sigma$) ↑ | Stability ↓ |
|---|---|---|---|---|---|---|
| Training from scratch (8-fold cross-validation) | | | | | | |
| U-Net | 0.854 ± 0.102 | 0.918 ± 0.059 | 0.921 ± 0.072 | 0.917 ± 0.063 | 0.973 ± 0.027 | 0.102 |
| U-Net-L | 0.738 ± 0.178 | 0.837 ± 0.121 | 0.813 ± 0.143 | 0.869 ± 0.113 | 0.953 ± 0.039 | 0.178 |
| UNet++ | 0.859 ± 0.113 | 0.920 ± 0.068 | 0.906 ± 0.101 | 0.940 ± 0.050 | 0.972 ± 0.033 | 0.113 |
| UNet++-L | 0.763 ± 0.165 | 0.855 ± 0.109 | 0.836 ± 0.132 | 0.882 ± 0.102 | 0.957 ± 0.038 | 0.165 |
| TransUNet | 0.865 ± 0.120 | 0.923 ± 0.073 | 0.940 ± 0.061 | 0.910 ± 0.099 | 0.978 ± 0.019 | 0.120 |
| PretrainedTransUNet[a] | **0.929 ± 0.064** | **0.962 ± 0.036** | **0.958 ± 0.037** | **0.965 ± 0.036** | **0.988 ± 0.011** | **0.064** |
| LightMUNet | 0.697 ± 0.182 | 0.808 ± 0.128 | 0.777 ± 0.156 | 0.851 ± 0.113 | 0.944 ± 0.046 | 0.182 |
| KITTI pretraining (8 fold cross-validation) | | | | | | |
| U-Net | **0.921 ± 0.064** | **0.958 ± 0.037** | **0.960 ± 0.035** | **0.956 ± 0.041** | **0.987 ± 0.011** | **0.064** |
| U-Net-L | 0.884 ± 0.089 | 0.936 ± 0.052 | 0.936 ± 0.044 | 0.937 ± 0.065 | 0.981 ± 0.016 | 0.089 |
| UNet++ | 0.917 ± 0.067 | 0.956 ± 0.038 | 0.957 ± 0.033 | 0.954 ± 0.044 | 0.986 ± 0.011 | 0.067 |
| UNet++-L | 0.815 ± 0.121 | 0.893 ± 0.076 | 0.881 ± 0.081 | 0.908 ± 0.081 | 0.968 ± 0.024 | 0.121 |
| TransUNet | 0.796 ± 0.136 | 0.880 ± 0.086 | 0.877 ± 0.123 | 0.895 ± 0.089 | 0.962 ± 0.036 | 0.136 |
| PretrainedTransUNet[a] | 0.899 ± 0.073 | 0.945 ± 0.042 | 0.943 ± 0.049 | 0.948 ± 0.044 | 0.983 ± 0.014 | 0.073 |
| LightMUNet | 0.578 ± 0.244 | 0.702 ± 0.198 | 0.647 ± 0.216 | 0.793 ± 0.204 | 0.918 ± 0.066 | 0.244 |

[a] PretrainedTransUNet denotes the model initialized with ImageNet-pretrained weights.

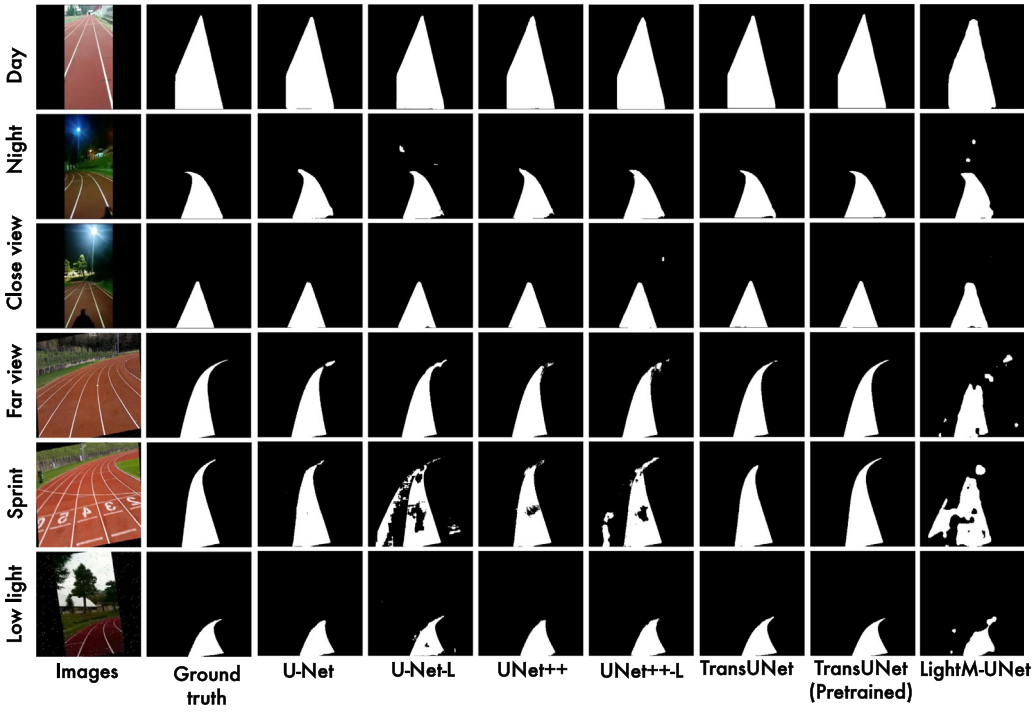



**Fig. 6.** Segmentation performance on the TrackAid-DT.

adverse conditions. U-Net and UNet++ maintain reliable segmentation in many cases, though thin lane markings are occasionally fragmented in difficult scenes. Lightweight variants exhibit reduced robustness when visual complexity increases. U-Net-L, UNet++-L, and LightM-UNet more frequently generate incomplete lane masks under low-contrast conditions, aligning with their lower IoU and recall values.

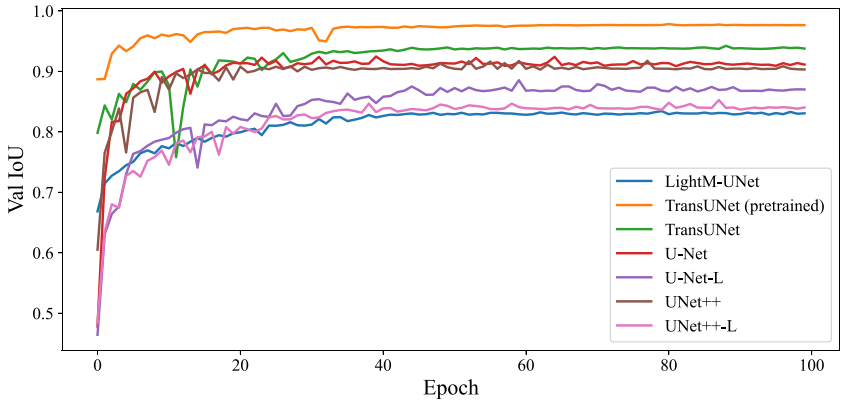
### 4.2. Carbon emissions

Energy and emissions tracking shows that training cost scales with model complexity. In benchmark studies, it is important to report energy and carbon costs to enable more advanced evaluations while remaining environmentally conscious, as shown in domains where continuous use is expected. The estimated training emissions scaled with model complexity, but this relationship was not always proportional to size of the model (Galdelli et al., 2025). The carbon emissions were measured over the complete training cycle encompassing all training epochs to report the total cumulative emissions for each model. The results are presented in Table 3, in the case of training from scratch, U-Net and U-Net-L had the lowest emissions (16.49 g $CO_2$e), UNet++ moderately higher (33.07 g $CO_2$e), UNet++L (23.43 g $CO_2$e), and

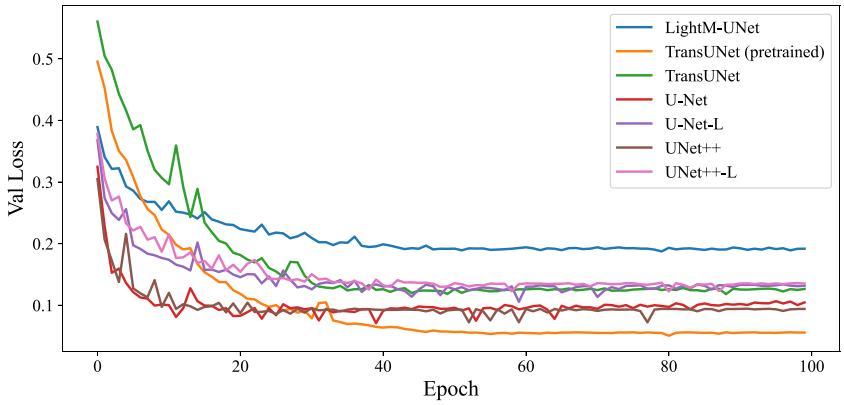
**Table 3**
Comparison of emissions for models trained from scratch and with KITTI pretraining along with runtime performance metrics.

| Model | Emissions [g $CO_2$e] ↓ | | Latency (ms) ↓ | FPS ↑ |
|---|---|---|---|---|
| | Scratch | KITTI pretraining | | |
| U-Net | **16.49** | 52.33 | **80.31** | **12.45** |
| U-Net-L | **16.49** | **51.09** | 84.32 | 11.86 |
| UNet++ | 33.07 | 89.01 | 85.11 | 11.75 |
| UNet++-L | 23.43 | 68.72 | 85.07 | 11.76 |
| TransUNet | 56.21 | 60.66 | 147.54 | 6.78 |
| PretrainedTransUNet[a] | 56.33 | 59.72 | 128.52 | 7.78 |
| LightMUNet | 130.38 | 550.97 | 150.91 | 6.63 |

[a] PretrainedTransUNet denotes the model initialized with ImageNet-pretrained weights.




(a) Validation IoU over epochs.




(b) Validation loss over epochs.

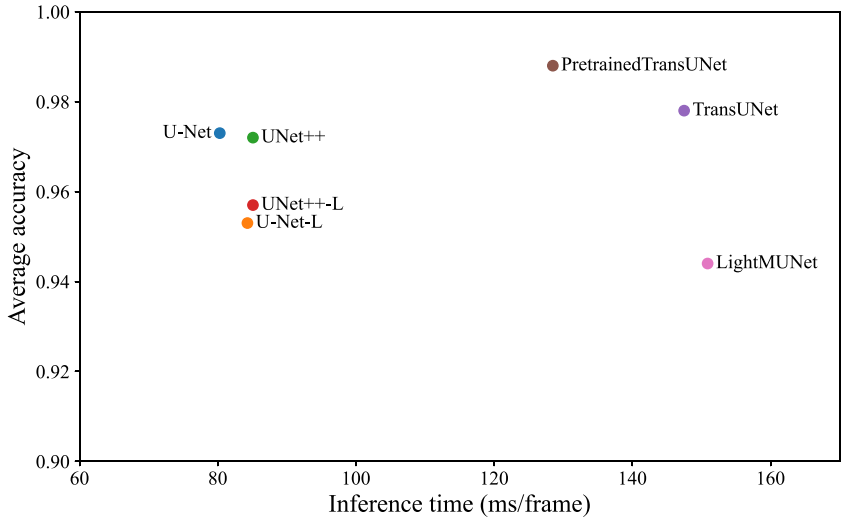
**Fig. 7.** Validation performance across architectures on KITTI dataset.

TransUNet variants both approximately around 56 g. Surprisingly, LightM-UNet reported the highest emissions (130.38 g $CO_2$e), despite being designed as a lightweight architecture. This outcome stems from the fact that LightM-UNet, while reducing parameter count, relies on a higher number of operations per layer and exhibits slower convergence during training. As a result, its overall compute time and energy consumption exceeded that of larger but more efficient architectures.

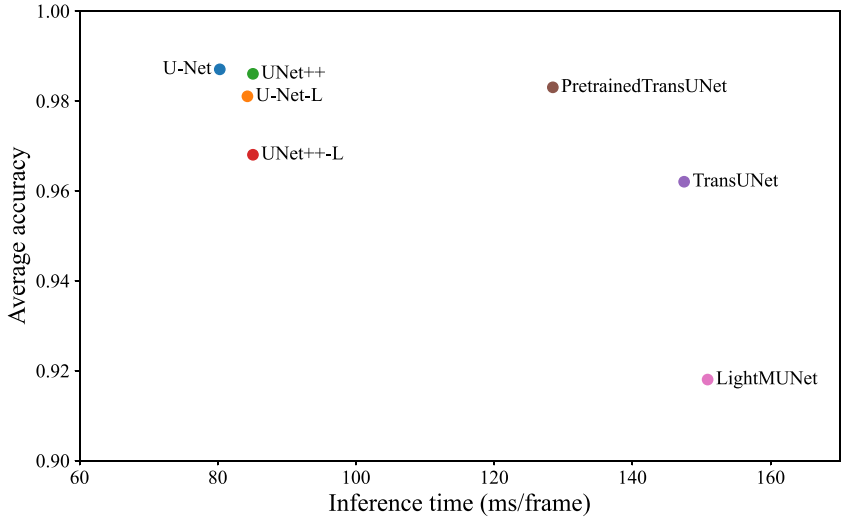
When models are pretrained on KITTI, overall training emissions increase across all architectures because of the added pretraining phase. The larger dataset size requires more training iterations and longer compute time, resulting in an increased GPU utilization and sustained power consumption over extended training durations, leading to increased carbon emissions. U-Net and U-Net-L remained among the most energy-efficient models, with average emissions of approximately 52.33 g and 51.09 g $CO_2$e, respectively. UNet++ and UNet++-L incurred moderately higher emissions of around 89.01 g and 68.72 g $CO_2$e, reflecting their increased architectural complexity. However, this does not lead to an improvement in segmentation performance. Transformer-based models again exhibited higher carbon costs, with TransUNet and its ImageNet pretrained variant reporting emissions close to 60.66 g and 59.72 g $CO_2$e. LightM-UNet showed a disproportionately high environmental cost, with emissions exceeding 550.97 g $CO_2$e. This behavior mirrors the trend observed in training from scratch and is attributed to slower convergence and increased computational overhead, despite the model's reduced parameter count. These findings demonstrate that architectural compactness does not guarantee lower environmental cost, underscoring the need to report emissions alongside accuracy and latency.

### 4.3. Training dynamics

Training dynamics were analyzed by inspecting validation IoU and validation loss curves across epochs for all architectures trained with KITTI pretraining. Fig. 7 summarize the evolution of validation performance and provide insight into convergence behavior, stability, and optimization efficiency. When trained from scratch, most architectures exhibited rapid improvements in validation IoU during the early epochs, followed by a gradual saturation phase. U-Net and UNet++ converged quickly and maintained stable IoU trajectories with low variance across epochs, indicating reliable optimization behavior. TransUNet achieved the highest validation IoU overall but showed larger fluctuations during training, including occasional instability in the loss curve, suggesting higher sensitivity to optimization dynamics. Lightweight variants such as LightM-UNet converged more slowly and saturated at lower IoU levels, reflecting reduced representational capacity and less robust learning behavior. Under KITTI pretraining, convergence was consistently faster across all architectures. Validation IoU increased sharply within the first few epochs, and loss values dropped more rapidly compared to training from scratch. Pretrained U-Net and UNet++ demonstrated smoother and more stable training curves, with reduced oscillations in both IoU and loss. Transformer-based models benefited noticeably from pretraining, showing improved stability and fewer abrupt deviations during optimization. This effect suggests that task specific pretrained representations provide a more favorable initialization, reducing sensitivity to learning rate and early training noise.




(a) Training from scratch.




(b) Pretraining with KITTI.

**Fig. 8.** Trade-off between accuracy and inference time (ms/frame) for different settings.

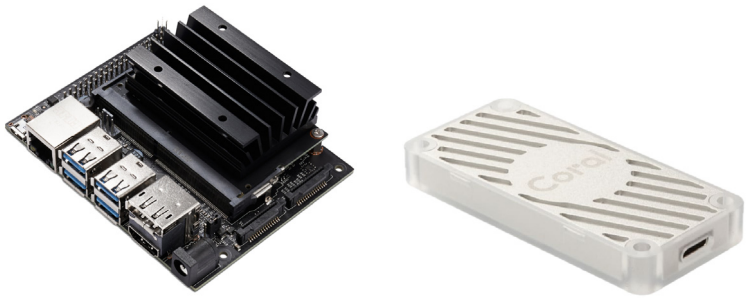
### 4.4. Runtime performance

Inference performance benchmarks on our workstation are illustrated in Fig. 8 and Table 3, which reports the relationship between average accuracy and inference latency for both training from scratch and KITTI pretraining. U-Net and UNet++ achieve inference times of approximately 80–84 ms per frame (around 12 FPS), offering a favorable balance between segmentation accuracy and runtime efficiency. TransUNet and TransUNet (initialized with ImageNet-pretrained weights) attains competitive accuracy but incurs higher latency at approximately 147 ms per frame (around 7 FPS) and 128 ms per frame (around 8 FPS), reflecting the additional computational cost introduced by transformer components. LightM-UNet reported the highest latency at 150.91 ms per frame (around 7 FPS) with less accuracy compared to other models. As a general trend, KITTI pretraining improves accuracy with increase in emissions across all architectures adding runtime overhead compared to models trained from scratch. Fig. 8 showcases that, convolutional models improve in terms of accuracy along with efficient inference behavior, whereas transformer based architectures remain constrained by higher runtime cost. Therefore, U-Net offers competitive trade-offs compared to lightweight variants, that reduce architectural complexity but suffer from a reduction in segmentation accuracy and do not consistently translate into lower inference latency.

### 4.5. Deployment on embedded devices

We selected U-Net for quantization and deployment, based on the evaluations of trade-offs in segmentation. U-Net offers a balance between accuracy and efficiency, that allows it to serve as a representative model in the lane detection application, for exploring deployment on constrained devices. We followed a progressive optimization pipeline to enable on-device inference on resource-constrained hardware. The most relevant tests were performed on the following edge devices:

- **NVIDIA® Jetson Nano**: The Jetson Nano (Fig. 9) is a development board produced by NVIDIA®, best known for its GPUs, and designed for AI applications with low power consumption (Narang et al., 2024). It integrates a quad-core ARM CPU running at up to 1.43 GHz, paired with a Maxwell GPU based on NVIDIA's Erista chip (Tegra X1) with 128 cores (half of a fully enabled Tegra X1). This configuration offers solid computational capability, particularly through GPU acceleration, with a typical power consumption of around 5 W.



(a) NVIDIA® Jetson Nano. (b) Google Coral USB Accelerator.

**Fig. 9.** Edge devices selected to benchmark inference performance on deployment.

**Table 4**
Deployment results summarizing accuracy preservation and latency gains.

| Model variant | Precision | Device | Latency (ms) | Accuracy |
|---|---|---|---|---|
| Reduced U-Net Lite | float32 (TFLite) | Jetson Nano CPU | ~100 | 0.981 |
| Reduced U-Net Lite Quantized | INT8 (TFLite) | Jetson Nano CPU | ~70 | 0.979 |
| Reduced U-Net Lite Quantized | INT8 (TFLite) | Coral Edge TPU | ~7 | 0.967 |

- **Google Coral USB Accelerator**: The Coral USB Accelerator (Fig. 9(b)) provides an accessible and portable way to leverage Google's Edge TPU technology for on-device inference (Google LLC., 2026). It is designed to connect via USB 3.0 to a host systems like a desktop, embedded board, or single-board computer like the NVIDIA® Jetson Nano and thereby enabling hardware accelerated execution of TensorFlow Lite models. It incorporates a dedicated Edge TPU coprocessor capable of up to 4 trillion operations per second (TOPS) at INT8 precision, while maintaining a remarkably low power consumption of approximately 0.5 W per TOPS.

Starting from a trained U-Net, the input resolution was first reduced to obtain a lighter baseline model. This reduction lowered the number of parameters and layers, decreased memory footprint, and reduced computational cost, while maintaining comparable segmentation quality. The reduced model was then converted into TensorFlow Lite (float32). This conversion produces a `.tflite` file that can be run with the TensorFlow Lite interpreter, optimized for ARM-based devices. This step reduces model size and enables embedded CPU deployment without retraining. As a further optimization, the TensorFlow Lite model was quantized to INT8 using post-training quantization. This process reduces the precision of weights and activations from 32-bit floating point to 8-bit integers. During conversion, the training set was used as a representative dataset to calibrate activation ranges. The resulting model is approximately four times smaller, executes integer arithmetic efficiently on ARM processors, and is fully compatible with the Coral Edge TPU. As shown in Table 4, quantization preserved accuracy within a negligible margin while significantly reducing inference latency. For practical purposes, the execution time was measured considering only the inference stage of the model, while excluding preprocessing operations such as image resizing. This assumption is justified by the fact that images acquired directly from the device camera can be expected to already conform to the required resolution and color scheme, thereby reducing overall execution time.

Finally, the quantized model was compiled with the Edge TPU compiler and deployed on the Coral Edge TPU accelerator connected to the Jetson Nano. In this configuration, all layers of the reduced U-Net quantized model were mapped to the TPU, enabling full hardware acceleration and real-time performance. Accuracy was preserved after both TensorFlow Lite conversion and quantization, with test accuracy remaining in the range of 0.97–0.99 with 0.98 as the average accuracy over the test set. Latency improved step by step: approximately 100 ms per frame for the Lite model on Jetson Nano CPU, 70 ms per frame for the quantized model on Jetson Nano CPU, and only 7 ms per frame on the Coral Edge TPU which is equivalent to about 140 frames per second.

### 4.6. Out-of-distribution generalization test

While the main dataset was restricted to track-mounted recordings to ensure a controlled and comparable benchmark across architectures, real deployment conditions, as noted by other studies (Mancini et al., 2018) are more diverse. To examine how well models extend beyond the training domain, we evaluated on additional contextual images not included in the original dataset representing our out of distribution test set. On top of the domain specific test set, an extra set of sixty images were incorporated to further evaluate the generalization capability of the proposed architectures. Unlike cross-validation, which assesses in-distribution variability, the generalization test exposes the models to new visual conditions, thereby highlighting differences in architectural inductive bias and transferability. The images are sourced from multiple public datasets (Karim, 2020; Kaundanyachinmaya07, 2024; Kpgeek, 2025; Silva, 2025; valentin-w91ey, 2025) to capture diverse conditions, and is intended as a targeted robustness check for more advanced studies. These images introduce greater diversity in terms of visual conditions and structural characteristics. This out-of-distribution generalization test enables a more rigorous assessment of model robustness. Representative examples of the additional images are provided in Fig. 10. These samples covered road scenes with lane markings of different geometry, frames with multiple athletes and partial occlusions, and environments with distracting painted surfaces. By keeping these cases separate, we avoid confounding the primary evaluation while still probing robustness under domain shift. Fig. 11, presents the results showing that models with pretrained encoders, such as TransUNet, exhibit more stable and coherent predictions when transferred to unseen contexts. In contrast, lighter variants display increased sensitivity to background clutter, resulting in fragmented segmentation, false positives, and missed detections. These qualitative differences underscore the importance of out-of-distribution generalization tests alongside standard benchmarks to better anticipate performance variability under real-world conditions. The quantitative results for lane segmentation on the generalization test are reported in Table 5, while the corresponding performance metrics for the background class are provided in the Table A.8. These results are presented alongside qualitative examples (in Fig. 11) to characterize model behavior under unseen conditions.

The generalization results indicate that the TransUNet (pretrained on ImageNet) achieves the best performance among all the tested models with an IoU of 0.580 and an F1 score of 0.734 on the lane

**Fig. 10.** Sample images from the out-of-distribution generalization test (Karim, 2020; Kaundanyachinmaya07, 2024; Kpgeek, 2025; Silva, 2025; valentin-w91ey, 2025).

**Table 5**
Results of models pretrained on KITTI and fine-tuned on TrackAid-DT, evaluated on the out-of-distribution generalization set.

| Architecture | IoU ↑ | F1 ↑ | Recall ↑ | Precision ↑ | Accuracy ↑ |
|---|---|---|---|---|---|
| U–Net | 0.404 | 0.576 | 0.448 | 0.807 | 0.848 |
| U–Net-L | 0.312 | 0.476 | 0.379 | 0.640 | 0.808 |
| UNet++ | 0.340 | 0.508 | 0.411 | 0.663 | 0.817 |
| UNet++–L | 0.321 | 0.486 | 0.408 | 0.603 | 0.802 |
| TransUNet | 0.521 | 0.685 | 0.553 | 0.900 | 0.882 |
| PretrainedTransUNet[a] | **0.580** | **0.734** | **0.594** | **0.959** | **0.900** |
| LightM–UNet | 0.292 | 0.452 | 0.362 | 0.599 | 0.798 |

[a] PretrainedTransUNet denotes the model initialized with ImageNet-pretrained weights.

class representing the area accessible to the athlete. These results indicate that combining convolutional feature extraction with transformer-based global context modeling, together with large-scale pretraining, enables the model to capture more transferable representations. The model also exhibits higher generalized precision, suggesting a strong ability to suppress false positives under domain shift conditions. TransUNet, trained from scratch, ranks second, confirming that transformer-based architectures can generalize effectively when sufficient structural capacity is available. However, the performance gap between TransUNet and its pretrained counterpart highlights the positive role of domain-relevant pretraining in improving robustness on unseen data. In contrast with hybrid convolution-transformer models, convolution-only encoder–decoder models such as U-Net and UNet++, which showed improved performance during cross-validation, experience a noticeable degradation in the generalization test. Although U-Net remains competitive, its lower recall indicates a conservative behavior, with a tendency to miss part of the region accessible to the athlete. This suggests that convolutional encoder–decoder models rely more on local texture statistics that are sensitive to dataset-specific characteristics. Lightweight and regularized architectures, including U-Net-L, UNet++-L, and LightM-UNet, exhibit a larger performance drop, with IoU values below 0.33.

**Table 6**
Human-in-the-loop evaluation results across outdoor trials, reporting mean speed and standard deviation computed over all runs (trials).

| Location | Number of runs | Average speed (m/s) | Std. Dev. (m/s) |
|---|---|---|---|
| Ancona | 10 | 2.59 | 0.20 |
| Fermo | 10 | 2.80 | 0.23 |
| Overall | 20 | 2.69 | 0.22 |

### 4.7. Outdoor experiments

This work includes a human-in-the-loop evaluation to assess whether the proposed perception and feedback pipeline supports continuous motion under real world conditions. The goal is feasibility verification at the system level, with emphasis on whether real time segmentation and haptic feedback remain usable during motion on an athletic track considering the regulations of sports events and standard training practices. Experiments were conducted on outdoor athletic tracks in the Marche region of central Italy, at two locations, Ancona and Fermo. The complete wearable system comprises a chest mounted camera, an embedded processing unit, and haptic gloves providing directional feedback. The total system weight is approximately 750 g and all the computation is performed on device, enabling real time inference and feedback without external connectivity. Sighted participants (2 users) wearing blindfolds were recruited to ensure safety while providing a controlled approximation of visual impairment, a protocol commonly used in assistive navigation research (X. Liu et al., 2024; Voutsakelis et al., 2025). Though, a study notes that there is always some difference in perception when blindfolding (Qiu et al., 2024), however for vibrotactile feedback related devices, the experience is more favorable for testing (Voutsakelis et al., 2025). During each trial, participants wore the full system and moved along the athletic track while remaining within the walkable track area. The participants performed complete loops of an athletic track (overall length of 400 m). Experiments were conducted on two separate days for each location under natural outdoor conditions without any controlled lighting setup. Each user performed 5 loops with a pause between each loop. The duration of each lap depended on individual's speed and was not externally constrained to focus on user experience.

Participants relied exclusively on haptic feedback to regulate direction and position. No visual or verbal cues were provided during motion. The system continuously processed camera input, performed segmentation of the accessible region, and translated the resulting guidance into haptic signals. Speed is used as a system level proxy for effectiveness because it integrates segmentation stability, inference latency, and feedback clarity into a single human centered outcome. The results in Table 6 show that participants were able to maintain consistent forward motion with overall speeds of 2.69 m per second, using haptic guidance alone, indicating that the system supports real time human use under realistic conditions. The reported standard deviation is computed over all runs (trials) at a location, pooling data from both participants at each location, and then finally averaged to indicate overall. The higher values of standard deviation indicate greater variability in speed across individual runs, while lower values indicate more consistent performance. More extensive evaluations involving visually impaired athletes under controlled protocols remain future work.

The results indicate that participants were able to maintain consistent forward motion, in running speeds, while relying exclusively on haptic feedback. Participants were instructed to remain within a designated lane of the track. While minor lateral deviations occurred, the haptic feedback enabled continuous correction, and users were able to remain within the intended track boundaries without external intervention. The interaction with the haptic feedback system indicated a short learning curve, allowing participants to quickly adapt to the guidance signals from the gloves. The main limitations of the current system is the absence of obstacle detection and multi-agent awareness,

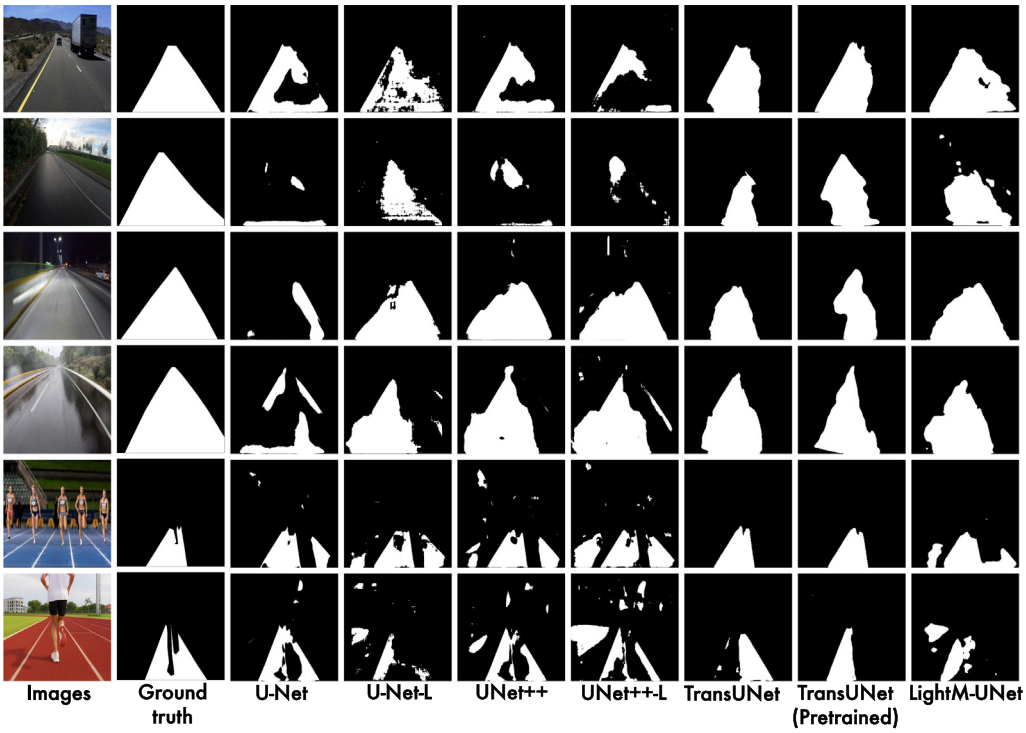



**Fig. 11.** Inference results on the out-of-distribution generalization test.

due to which it is useful for practice sessions. Another limitations is that the current system is tested only for few training locations. Due to this, the system assumes a relatively structured environment (track boundaries), and performance may degrade in more visually complex or cluttered scenes. Despite this, for trajectories with moderate to large curvature radius, the controller reliably guided users along the intended path. Average speeds remain stable across users, with low variance between runs, suggesting that segmentation output, inference latency, and feedback timing are sufficiently stable to avoid hesitation or corrective stopping. The observed speeds fall within a controlled range, which is appropriate for blindfolded operation and confirms that the real-time perception pipeline supports continuous human motion rather than intermittent or stop and go behavior, and with better training improves. Variability across participants remains limited, indicating that system performance dominates over individual adaptation effects in this setting. The current challenging factors of the outdoor experiments may lead to occasional deviations or delayed corrections in more complicated test scenes, which should be addressed in systematic testing in future through the use of the system.

## 5. Discussion

### 5.1. Key insights and implications for assistive sports

The experimental results reveal clear trade-offs between the more accurate models and those that are most practical for deployment on edge environments. Transformer-based architectures such as TransUNet achieved state-of-the-art segmentation accuracy, but their higher latency and elevated training emissions make them challenging to use in real-time, resource-constrained environments. This could be potentially considered especially in other application of the practical use cases, such as indoor athletic activities with less network constraints to conduct frame processing remotely

This constraint is particularly important in the context of guiding visually impaired athletes, where timely lane detection is critical. In such applications, reliable feedback matters more than marginal improvements in accuracy, as delays can hinder an athlete's ability to adjust movement and maintain orientation. Lightweight variants went to the opposite extreme: they reduced computational load but struggled to maintain reliable lane continuity, especially under low-light conditions. In our application, U-Net struck the most effective balance. It combined high accuracy with the lowest emissions and competitive inference speed, and once quantized, it maintained performance while running at real-time frame rates on embedded devices. For assistive sports applications, where timely feedback and portability are critical, these characteristics outweigh marginal gains in accuracy from larger transformer-based architectures.

In the case of the limitation of dataset size of our study, we employ cross-validation and pretraining strategies, and our findings indicate that the effectiveness of pretraining is architecture-dependent. Convolutional encoder decoder models benefit substantially from transfer learning, achieving both higher accuracy and improved stability, whereas transformer-based models appear to require either domain-specific pretraining or sufficient target-domain data to fully exploit their modeling capacity. These observations highlight the importance of jointly considering mean performance and variance when selecting segmentation models for real-world applications, particularly in scenarios characterized by limited domain overlap.

Considering the out-of-distribution generalization test, the evaluated models demonstrate effectiveness under constrained computational settings; however, architectures with limited representational capacity struggle to accommodate the variability present in the external dataset. It is important to note, however, that the external generalization dataset, while useful, is not fully representative of the athlete-specific egocentric scenario motivating this work. Several test instances correspond to road environments with lane markings visible only on one side with geometries that differ substantially from standard athletic tracks. This demonstrates the usefulness of TrackAid-DT, which encodes the visual structure of track walkable areas. Compared to prior approach that relies on classical computer vision pipelines without any learning from the dataset (Mancini et al., 2018), the learning-based models evaluated in this study capture more transferable spatial and structural cues. This is further reflected in the prototype evaluations in the Section 4.7, where the proposed vision-based pipeline provides

stable and usable guidance at higher speed under real conditions. These findings support the relevance of learning-based egocentric perception for assistive athletic applications, while also highlighting the need for future generalization and specific datasets that closely reflect athlete-specific environments.

Much of the prior literature on lane detection has prioritized maximizing accuracy through deeper backbones or transformer-based encoders. Our study extends this perspective by treating efficiency and sustainability as first-class metrics, alongside segmentation quality. In doing so, we show that a carefully optimized U-Net can rival or surpass more complex designs when these dimensions of performance are considered. This reframing is important: it demonstrates that legacy architectures remain competitive when adapted for embedded deployment, suggesting that progress in this field should be evaluated not just by accuracy benchmarks, but also by deployment viability in real-world conditions.

### 5.2. Limitations

Our system has shown promising results for helping visually impaired athletes with training activities at running speeds. However, a few limitations warrant consideration. First, our primary dataset was collected in controlled athletic track environments that are suitable for training activities, yet it may not capture the full variability encountered in real-world sports events. Although we employed established strategies such as session-wise cross-validation, external pretraining, and out-of-distribution generalization test to mitigate data limitations; future work should extend data collection to a broader range of environments, including different stadium layouts, surface materials, and camera perspectives. Second, deployment benchmarks were limited to the NVIDIA Jetson Nano platform, evaluated both natively and with an attached Coral Edge TPU accelerator. While this configuration reflects a representative wearable edge scenario, exploring a wider range of embedded hardware, including mobile and consumer-grade devices, would provide a more comprehensive assessment of deployment readiness. Third, the reported emission estimates are coarse-grained and based on aggregate measurements. Integrating fine-grained, hardware-level power monitoring would enable more accurate energy profiling and deeper insights into the sustainability trade-offs of embedded inference. Finally, the proposed end-to-end system represents an initial prototype for vision-based assistive running. The current evaluation does not account for scenarios involving multiple runners, dynamic obstacles, or crowded environments. As such, the system should presently be regarded as a guidance and training aid (for instance in practice sessions) rather than a fully autonomous solution for unconstrained competitive settings. Extending the perception pipeline to jointly detect obstacles, along with outdoor experiments including other athletes, and contextual cues during running constitutes an important direction for future work.

## 6. Conclusion and future work

This work investigated fully autonomous running assistance for visually impaired athletes and proposed an end-to-end vision-based guidance system based on egocentric perception from a chest-mounted camera and real-time vibrotactile feedback delivered through wearable gloves. Built upon the TrackAid-DT, the proposed system addresses a task-specific and previously underexplored scenario. To the best of our knowledge, this is the first approach in the literature to support safe athletic running rather than walking or jogging under real-world conditions. TrackAid-DT comprises egocentric, pixel-annotated monocular images acquired under diverse orientations, speeds, and lighting conditions, and enables systematic training and evaluation of modern segmentation architectures. We conducted a comprehensive benchmark using an 8-fold cross-validation protocol from scratch and task specific pretraining with KITTI public dataset and report segmentation accuracy together with computational efficiency on both training and out-of-distribution generalization sets. The complete pipeline was further validated through deployment on multiple embedded edge devices and outdoor prototype testing at two locations, demonstrating its practical feasibility in realistic running scenarios. The study followed a consistent and deployment-oriented experimental protocol, combining lightweight preprocessing, a unified training setup across architectures, and an evaluation suite that jointly considers segmentation quality, inference latency, throughput, and $CO_2$-equivalent emissions. The results highlight clear trade-offs, where, higher-capacity models achieve stronger segmentation performance at increased computational and energy cost, whereas lightweight architectures offer competitive accuracy with substantially lower latency and emissions. Measuring training-related emissions provided additional insight into the environmental impact associated with different modeling choices. The evaluations in outdoor experiments indicate that the proposed system enables runners to maintain consistent forward motion relying exclusively on vibrotactile guidance, achieving higher running speeds compared to baseline studies. The performance variability observed across participants remained limited, suggesting that system behavior dominates over individual adaptation effects in the evaluated setting. Future work will focus on incorporating temporal modeling for sequence-aware inference, improving robustness through probability calibration and lightweight ensembling, and exploring quantization-aware training and mobile deployment platforms. In addition, extending the perception pipeline beyond lane walkable-area segmentation to jointly detect dynamic obstacles, other athletes, and relevant contextual cues during running represents a crucial step toward safer and more comprehensive assistive guidance in unconstrained environments.

## CRediT authorship contribution statement

**Gagan Narang:** Writing – review & editing, Writing – original draft, Visualization, Software, Methodology, Data curation, Conceptualization. **Alessandro Galdelli:** Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Oleksandr Kuznetsov:** Writing – review & editing, Validation, Supervision, Conceptualization. **Primo Zingaretti:** Writing – review & editing, Supervision, Resources, Funding acquisition. **Adriano Mancini:** Writing – review & editing, Validation, Supervision, Software, Project administration, Data curation, Conceptualization.

## Declaration of Generative AI and AI-assisted technologies in the writing process

During the preparation of this work the authors used the tool Grammarly in order to check grammar and spelling. After using the tool, the authors reviewed and edited the content as needed and take full responsibility for the content of the published article.

## Funding sources

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgments

The authors gratefully acknowledge the valuable contributions of master's students Tribuiani Lorenzo, Angelo Kollcaku and Joshua Sgariglia for their support during the development of this work.

**Table A.7**
Model performance under cross-validation training from scratch and KITTI pretraining, reported for background (bg) and foreground (fg) classes.

| Model | IoU (bg) ↑ | IoU (fg) ↑ | F1 (bg) ↑ | F1 (fg) ↑ | Rec. (bg) ↑ | Rec. (fg) ↑ | Prec. (bg) ↑ | Prec. (fg) ↑ | Stab. (bg) ↓ | Stab. (fg) ↓ |
|---|---|---|---|---|---|---|---|---|---|---|
| Training from scratch (8-fold cross-validation) | | | | | | | | | | |
| U–Net | 0.966 ± 0.035 | 0.854 ± 0.102 | 0.982 ± 0.018 | 0.918 ± 0.059 | 0.986 ± 0.010 | 0.921 ± 0.072 | 0.979 ± 0.031 | 0.917 ± 0.063 | 0.035 | 0.102 |
| U–Net–L | 0.945 ± 0.047 | 0.738 ± 0.178 | 0.971 ± 0.025 | 0.837 ± 0.121 | 0.980 ± 0.016 | 0.813 ± 0.143 | 0.963 ± 0.041 | 0.869 ± 0.113 | 0.047 | 0.178 |
| UNet++ | 0.966 ± 0.042 | 0.859 ± 0.113 | 0.982 ± 0.022 | 0.920 ± 0.068 | 0.990 ± 0.007 | 0.906 ± 0.101 | 0.975 ± 0.041 | 0.940 ± 0.050 | 0.042 | 0.113 |
| UNet++–L | 0.949 ± 0.046 | 0.763 ± 0.165 | 0.973 ± 0.024 | 0.855 ± 0.109 | 0.982 ± 0.014 | 0.836 ± 0.132 | 0.965 ± 0.042 | 0.882 ± 0.102 | 0.046 | 0.165 |
| TransUNet | 0.973 ± 0.021 | 0.865 ± 0.120 | 0.986 ± 0.011 | 0.923 ± 0.073 | 0.986 ± 0.012 | 0.940 ± 0.061 | 0.987 ± 0.014 | 0.910 ± 0.099 | 0.021 | 0.120 |
| PretrainedTransUNet[a] | **0.985 ± 0.013** | **0.929 ± 0.064** | **0.992 ± 0.006** | **0.962 ± 0.036** | **0.994 ± 0.005** | **0.958 ± 0.037** | **0.990 ± 0.010** | **0.965 ± 0.036** | **0.013** | **0.064** |
| LightM–UNet | 0.935 ± 0.055 | 0.697 ± 0.182 | 0.965 ± 0.030 | 0.808 ± 0.128 | 0.979 ± 0.014 | 0.777 ± 0.156 | 0.954 ± 0.051 | 0.851 ± 0.113 | 0.055 | 0.182 |
| KITTI pretraining (8-fold cross-validation) | | | | | | | | | | |
| U–Net | **0.984 ± 0.013** | **0.921 ± 0.064** | **0.992 ± 0.006** | **0.958 ± 0.037** | **0.992 ± 0.006** | **0.960 ± 0.035** | **0.992 ± 0.008** | **0.956 ± 0.041** | **0.013** | **0.064** |
| U–Net–L | 0.977 ± 0.019 | 0.884 ± 0.089 | 0.989 ± 0.010 | 0.936 ± 0.052 | 0.989 ± 0.011 | 0.936 ± 0.044 | 0.987 ± 0.012 | 0.937 ± 0.065 | 0.019 | 0.089 |
| UNet++ | 0.983 ± 0.013 | 0.917 ± 0.067 | 0.991 ± 0.006 | 0.956 ± 0.038 | 0.991 ± 0.007 | 0.957 ± 0.033 | **0.992 ± 0.007** | 0.954 ± 0.044 | **0.013** | 0.067 |
| UNet++–L | 0.962 ± 0.030 | 0.815 ± 0.121 | 0.980 ± 0.016 | 0.893 ± 0.076 | 0.986 ± 0.012 | 0.881 ± 0.081 | 0.975 ± 0.024 | 0.908 ± 0.081 | 0.030 | 0.121 |
| TransUNet | 0.954 ± 0.045 | 0.796 ± 0.136 | 0.976 ± 0.024 | 0.880 ± 0.086 | 0.985 ± 0.011 | 0.877 ± 0.123 | 0.969 ± 0.045 | 0.895 ± 0.089 | 0.045 | 0.136 |
| PretrainedTransUNet[a] | 0.978 ± 0.018 | 0.899 ± 0.073 | 0.989 ± 0.009 | 0.945 ± 0.042 | 0.991 ± 0.006 | 0.943 ± 0.049 | 0.986 ± 0.017 | 0.948 ± 0.044 | 0.018 | 0.073 |
| LightM–UNet | 0.908 ± 0.073 | 0.578 ± 0.244 | 0.950 ± 0.041 | 0.702 ± 0.198 | 0.971 ± 0.032 | 0.647 ± 0.216 | 0.933 ± 0.063 | 0.793 ± 0.204 | 0.073 | 0.244 |

**bg** = background, **fg** = foreground.
[a] PretrainedTransUNet denotes the model initialized with ImageNet-pretrained weights.

**Table A.8**
Results of models pretrained on KITTI and fine-tuned on TrackAid-DT, evaluated on the out-of-distribution generalization set.

| Architecture | IoU (bg) ↑ | IoU (fg) ↑ | F1 (bg) ↑ | F1 (fg) ↑ | Rec. (bg) ↑ | Rec. (fg) ↑ | Prec. (bg) ↑ | Prec. (fg) ↑ | Acc. ↑ |
|---|---|---|---|---|---|---|---|---|---|
| U–Net | 0.830 | 0.404 | 0.907 | 0.576 | 0.992 | 0.448 | 0.836 | 0.807 | 0.848 |
| U–Net-L | 0.790 | 0.312 | 0.883 | 0.476 | 0.937 | 0.379 | 0.835 | 0.640 | 0.808 |
| UNet++ | 0.798 | 0.340 | 0.887 | 0.508 | 0.938 | 0.411 | 0.842 | 0.663 | 0.817 |
| UNet++–L | 0.782 | 0.321 | 0.877 | 0.486 | 0.920 | 0.408 | 0.839 | 0.603 | 0.802 |
| TransUNet | 0.865 | 0.521 | 0.928 | 0.685 | 0.981 | 0.553 | 0.879 | 0.900 | 0.882 |
| PretrainedTransUNet[a] | **0.884** | **0.580** | **0.939** | **0.734** | **0.992** | **0.594** | **0.890** | **0.959** | **0.900** |
| LightM–UNet | 0.779 | 0.292 | 0.876 | 0.452 | 0.928 | 0.362 | 0.830 | 0.599 | 0.798 |

**bg** = background, **fg** = foreground.
[a] PretrainedTransUNet denotes the model initialized with ImageNet-pretrained weights.

## Appendix. Additional quantitative results

In addition to overall accuracy, we compute class-wise metrics to distinguish between lane foreground (fg) and lane background (bg) performance, as reported in Table A.7 and Table A.8. Foreground accuracy measures the proportion of lane pixels correctly identified, while background accuracy measures the proportion of non-lane pixels correctly identified.

## Data and software availability

The code and the dataset TrackAid-DT for the reproducibility is publicly released at https://github.com/vrai-group/TrackAid-DT.

## References


Afaq, Y., Akram, S.V., 2025. Integration of deep learning with edge computing on progression of societal innovation in smart city infrastructure: A sustainability perspective. Sustain. Futur. 9, 100761. http://dx.doi.org/10.1016/j.sftr.2025.100761.

Alfano, P.D., Pastore, V.P., Rosasco, L., Odone, F., 2024. Top-tuning: A study on transfer learning for an efficient alternative to fine tuning for image classification with fast kernel methods. Image Vis. Comput. 142, 104894.

Åström, K.J., Murray, R.M., 2008. Feedback Systems: An Introduction for Scientists and Engineers. Princeton University Press.

Behrendt, K., Soussan, R., 2019. Unsupervised labeled lane markers using maps. In: 2019 IEEE/CVF International Conference on Computer Vision Workshop. ICCVW, pp. 832–839. http://dx.doi.org/10.1109/ICCVW.2019.00111.

Biancucci, M., Galdelli, A., Narang, G., Pietrini, R., Mancini, A., Zingaretti, P., 2025. COIGAN: Controllable Object Inpainting Through Generative Adversarial Network for Defect Synthesis in Data Augmentation. In: 2025 IEEE International Conference on Robotics and Automation. ICRA, pp. 9046–9052. http://dx.doi.org/10.1109/ICRA55743.2025.11127765.

Chen, J., Lu, Y., Yu, Q., Luo, X., Adeli, E., Wang, Y., Lu, L., Yuille, A.L., Zhou, Y., 2021. TransUNet: Transformers Make Strong Encoders for Medical Image Segmentation. arXiv:2102.04306 URL: https://arxiv.org/abs/2102.04306.

Chen, J., Mei, J., Li, X., Lu, Y., Yu, Q., Wei, Q., Luo, X., Xie, Y., Adeli, E., Wang, Y., Lungren, M.P., Zhang, S., Xing, L., Lu, L., Yuille, A., Zhou, Y., 2024. TransUNet: Rethinking the U-Net architecture design for medical image segmentation through the lens of transformers. Med. Image Anal. 97, 103280. http://dx.doi.org/10.1016/j.media.2024.103280.

Davies, T., 2012. Visually impaired athletes competing in the olympic stadium. URL: https://commons.wikimedia.org/wiki/File:Visually_impaired_athletes_competing_in_the_Olympic_Stadium_(9375674645).jpg Creative Common License 2.0 (https://creativecommons.org/licenses/by/2.0/) Original file. (Accessed: 15 April 2026) Wikimedia Commons.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., Fei-Fei, L., 2009. ImageNet: A large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition. pp. 248–255. http://dx.doi.org/10.1109/CVPR.2009.5206848.

Fritsch, J., Kühnl, T., Geiger, A., 2013. A new performance measure and evaluation benchmark for road detection algorithms. In: 16th International Conference on Intelligent Transportation Systems. ITSC 2013, pp. 1693–1700. http://dx.doi.org/10.1109/ITSC.2013.6728473.

Galdelli, A., Narang, G., Pietrini, R., Zazzarini, M., Fiorani, A., Tassetti, A.N., 2025. Multimodal AI-enhanced ship detection for mapping fishing vessels and informing on suspicious activities. Pattern Recognit. Lett. 191, 15–22. http://dx.doi.org/10.1016/j.patrec.2025.02.022.

Geiger, A., Lenz, P., Urtasun, R., 2012. Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. CVPR, URL: https://www.cvlibs.net/publications/Geiger2012CVPR.pdf (Accessed: 15 April 2026).

Google LLC., 2026. Coral USB Accelerator. URL: https://coral.ai/products/accelerator (Accessed: 15 April 2026).

Han, Z., Gu, J., Feng, Y., 2023. Blind lane detection and following for assistive navigation of vision impaired people. In: 2023 International Conference on Advanced Robotics and Mechatronics. ICARM, IEEE, pp. 721–726.

Han, S., Mao, H., Dally, W.J., 2016. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. URL: https://arxiv.org/abs/1510.00149 Oral presentation.

Han, K., Wang, Y., Tian, Q., Guo, J., Xu, C., Xu, C., 2020. GhostNet: More features from cheap operations. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition. CVPR, pp. 1577–1586. http://dx.doi.org/10.1109/CVPR42600.2020.00165.

Hou, Y., Ma, Z., Liu, C., Loy, C.C., 2019. Learning Lightweight Lane Detection CNNs by Self Attention Distillation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. ICCV, pp. 1013–1021. http://dx.doi.org/10.1109/ICCV.2019.00110.

Howard, A., Sandler, M., Chu, G., Chen, L.-C., Chen, B., Tan, M., Wang, W., Tran, Y., Vasudevan, V., Adam, H., Le, Q., 2019. Searching for MobileNetV3. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. ICCV, pp. 1314–1324. http://dx.doi.org/10.1109/ICCV.2019.00140.

Jacob, B., Kligys, S., Chen, B., Zhu, M., Tang, M., Howard, A., Adam, H., Kalenichenko, D., 2018. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. CVPR, pp. 2704–2713.

Karim, F., 2020. Clean/dirty road classification. https://www.kaggle.com/datasets/faizalkarim/cleandirty-road-classification (Accessed: 15 April 2026).

Kaundanyachinmaya07, C., 2024. Micromobility lane recognition dataset. https://www.kaggle.com/datasets/kaundanyachinmaya07/micromobility-lane-recognition-dataset (Accessed: 15 April 2026).

Kemeth, F., Hafenecker, S., Jakab, Á., Varga, M., Csielka, T., Couronné, S., 2014. BLINDTRACK: Guiding System for Visually Impaired - Locating System for Running on a Track. In: Proceedings of the 2nd International Congress on Sports Sciences Research and Technology Support. SciTePress, pp. 183–189. http://dx.doi.org/10.5220/0005143001830189.

Khalil, H.K., 2002. Nonlinear Systems. Prentice Hall, Upper Saddle River, N.J..

Kim, H., Kang, I., Baek, D., Jung, H., Morales, C., 2024. Lightweight semantic segmentation model for disaster area based on TransUNet. In: 2024 15th International Conference on Information and Communication Technology Convergence. ICTC, pp. 289–294. http://dx.doi.org/10.1109/ICTC62082.2024.10826955.

Kpgeek, 2025. KITTI road/lane detection dataset (224 ×224). URL: https://www.kaggle.com/datasets/kpgeek/kitti-roadlane-detection-dataset-224-x-224 (Accessed: 15 April 2026) Kaggle dataset.

Lacoste, A., Luccioni, A.S., Schmidt, V., Dandres, T., 2019. CodeCarbon: Estimate the carbon footprint of your computing. URL: https://github.com/mlco2/codecarbon (Accessed: 15 April 2026).

Leo, M., Medioni, G., Trivedi, M., Kanade, T., Farinella, G., 2017. Computer vision for assistive technologies. Comput. Vis. Image Underst. 154, 1–15. http://dx.doi.org/10.1016/j.cviu.2016.09.001.

Li, H., Xiong, P., Fan, H., Sun, J., 2019. DFANet: Deep Feature Aggregation for Real-Time Semantic Segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9514–9523. http://dx.doi.org/10.1109/CVPR.2019.00975.

Liao, W., Zhu, Y., Wang, X., Pan, C., Wang, Y., Ma, L., 2024. LightM-UNet: Mamba assists in lightweight UNet for medical image segmentation. arXiv:2403.05246 URL: https://arxiv.org/abs/2403.05246 (Accessed: 15 April 2026).

Liu, H., Rivera Soto, R.A., Xiao, F., Jae Lee, Y., 2021. YolactEdge: Real-time instance segmentation on the edge. In: 2021 IEEE International Conference on Robotics and Automation. ICRA, pp. 9579–9585. http://dx.doi.org/10.1109/ICRA48506.2021.9561858.

Liu, Y., Tian, Y., Zhao, Y., Yu, H., Xie, L., Wang, Y., Ye, Q., Jiao, J., Liu, Y., 2024. Vmamba: Visual state space model. In: Advances in Neural Information Processing Systems. Vol. 37, Curran Associates, Inc., pp. 103031–103063, URL: https://proceedings.neurips.cc/paper_files/paper/2024/file/baa2da9ae4bfed26520bb61d259a3653-Paper-Conference.pdf (Accessed: 15 April 2026).

Liu, X., Wang, B., Li, Z., 2024. Vision-based wearable steering assistance for people with impaired vision in jogging. In: 2024 IEEE International Conference on Robotics and Automation. ICRA, pp. 15270–15275. http://dx.doi.org/10.1109/ICRA57147.2024.10610660.

Mancini, A., Frontoni, E., Zingaretti, P., 2018. Mechatronic system to help visually impaired users during walking and running. IEEE Trans. Intell. Transp. Syst. 19 (2), 649–660. http://dx.doi.org/10.1109/TITS.2017.2780621.

Marques, M.P., Alves, A.C.d., 2021. Investigating environmental factors and paralympic sports: an analytical study. Disabil. Rehabil.: Assist. Technol. 16 (4), 414–419. http://dx.doi.org/10.1080/17483107.2020.1780483.

Narang, G., Berardini, D., Pietrini, R., Tassetti, A.N., Mancini, A., Galdelli, A., 2024. Edge-AI for buoy detection and mussel farming: a comparative study of YOLO frameworks. In: 2024 20th IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications. MESA, IEEE, pp. 1–8. http://dx.doi.org/10.1109/MESA61532.2024.10704814.

Neven, D., De Brabandere, B., Georgoulis, S., Proesmans, M., Van Gool, L., 2018. Towards end-to-end lane detection: an instance segmentation approach. In: 2018 IEEE Intelligent Vehicles Symposium. IV, pp. 286–291. http://dx.doi.org/10.1109/IVS.2018.8500547.

Pan, X., Shi, J., Luo, P., Wang, X., Tang, X., 2018. Spatial As Deep: Spatial CNN for Traffic Scene Understanding. In: Proceedings of the AAAI Conference on Artificial Intelligence. pp. 7276–7283. http://dx.doi.org/10.1609/aaai.v32i1.12301.

Patra, S., Maheshwari, P., Yadav, S., Banerjee, S., Arora, C., 2018. A joint 3d-2d based method for free space detection on roads. In: 2018 IEEE Winter Conference on Applications of Computer Vision. WACV, IEEE, pp. 643–652.

Pieralisi, M., Petrini, V., Di Mattia, V., Manfredi, G., De Leo, A., Scalise, L., Russo, P., Cerri, G., 2015. Design and realization of an electromagnetic guiding system for blind running athletes. Sensors 15 (7), 16466–16483. http://dx.doi.org/10.3390/s150716466.

Qin, Z., Wang, H., Li, X., 2020. Ultra fast structure-aware deep lane detection. In: European Conference on Computer Vision. Springer, pp. 276–291, https://www.ecva.net/papers/eccv_2020/papers_ECCV/papers/123690273.pdf.

Qiu, S., Hu, J., Han, T., Rauterberg, M., 2024. Can blindfolded users replace blind ones in product testing? an empirical study. Behav. Inf. Technol. 43 (8), 1664–1682.

Ronneberger, O., Fischer, P., Brox, T., 2015. U-Net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, pp. 234–241.

Silva, S.L.Á., 2025. Lane detection dataset morelos. URL: https://www.kaggle.com/datasets/sergiolvarezsilva/lane-detection-dataset-morelos-sergio (Accessed: 15 April 2026) Kaggle dataset.

Tabelini, L., Berriel, R., Paixão, T.M., Badue, C., De Souza, A.F., Oliveira-Santos, T., 2021a. Keep your eyes on the lane: Real-time attention-guided lane detection. In: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition. CVPR, pp. 294–302. http://dx.doi.org/10.1109/CVPR46437.2021.00036.

Tabelini, L., Berriel, R., Paixão, T.M., Badue, C., De Souza, A.F., Oliveira-Santos, T., 2021b. PolyLaneNet: Lane estimation via deep polynomial regression. In: 2020 25th International Conference on Pattern Recognition. ICPR, pp. 6150–6156. http://dx.doi.org/10.1109/ICPR48806.2021.9412265.

valentin-w91ey, 2025. Sprinter Athletes Tracking. Roboflow Universe dataset, URL: https://universe.roboflow.com/valentin-w91ey/sprinter-athletes-tracking (Accessed: 15 April 2026).

Voutsakelis, G., Dimkaros, I., Tzimos, N., Kokkonis, G., Kontogiannis, S., 2025. Development and evaluation of a tool for blind users utilizing ai object detection and haptic feedback. Machines 13 (5), 398.

Wen, X., Famouri, M., Hryniowski, A., Wong, A., 2021. AttendSeg: A Tiny Attention Condenser Neural Network for Semantic Segmentation on the Edge. arXiv:2104.14623 URL: https://arxiv.org/abs/2104.14623 15 April 2026.

Wing, M.K., Deuville, J., Grimes, A.C., Scanlan, Z., Arbour-Nicitopoulos, K., Latimer-Cheung, A.E., 2025. The sport experiences of blind or partially sighted people and strategies to support their participation in sport: A scoping review. Br. J. Vis. Impair. 02646196251330155, URL: https://doi.org/10.1177/02646196251330155.

World Athletics, 2024. World athletics competition and technical rules. (Accessed: 22 January 2026) https://worldathletics.org/about-iaaf/documents/book-of-rules.

World Para Athletics, 2024. World para athletics rules and regulations. (Accessed: 22 January 2026) https://www.paralympic.org/athletics/rules.

Wu, D., Liao, M., Zhang, W., Wang, X., Bai, X., Cheng, W., Liu, W., 2022. YOLOP: You only look once for panoptic driving perception. Mach. Intell. Res. 19 (6), 550–562. http://dx.doi.org/10.1007/s11633-022-1339-y.

Yao, W., Bai, J., Liao, W., Chen, Y., Liu, M., Xie, Y., 2024. From cnn to transformer: A review of medical image segmentation models. J. Imaging Inform. Med. 37 (4), 1529–1547.

Yu, F., Chen, H., Wang, X., Xian, W., Chen, Y., Liu, F., Madhavan, V., Darrell, T., 2020. BDD100K: A Diverse Driving Dataset for Heterogeneous Multitask Learning. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition. CVPR, pp. 2633–2642. http://dx.doi.org/10.1109/CVPR42600.2020.00271.

Yu, C., Wang, J., Peng, C., Gao, C., Yu, G., Sang, N., 2018. BiSeNet: Bilateral Segmentation Network for Real-Time Semantic Segmentation. In: Proceedings of the European Conference on Computer Vision. ECCV, http://dx.doi.org/10.1007/978-3-030-01261-8_20.

Zhang, S., Jia, Y., Liu, Z., Yang, F., 2024. Lightweight TransUNet Informal Residential Area segmentation. In: 2024 2nd International Conference on Machine Vision, Image Processing & Imaging Technology. MVIPIT, IEEE, pp. 90–96.

Zheng, T., Feng, Y., Xu, Y., Xu, Q., Zhong, Z., Lin, Z., Ren, J., Ma, L., 2021. RESA: Recurrent feature-shift aggregator for lane detection. In: Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 35, pp. 3547–3554. http://dx.doi.org/10.1609/aaai.v35i4.16469.

Zhou, Z., Rahman Siddiquee, M.M., Tajbakhsh, N., Liang, J., 2018. UNet++: A nested U-Net architecture for medical image segmentation. In: International Workshop on Deep Learning in Medical Image Analysis. Springer, pp. 3–11.

Zhou, Z., Siddiquee, M.M.R., Tajbakhsh, N., Liang, J., 2020. UNet++: Redesigning skip connections to exploit multiscale features in image segmentation. IEEE Trans. Med. Imaging 39 (6), 1856–1867. http://dx.doi.org/10.1109/TMI.2019.2959609.