УДК 378:004
DOI: 10.25686/978-5-8158-2474-4-2025-215-226

Исследование методов отслеживания зрачков для систем автономного вождения

Ч. Ван, Д. Д. Жданов, А. Д. Жданов
Университет ИТМО, Санкт-Петербург, Россия

Аннотация. В ответ на сложную задачу прогнозирования намерений пешеходов в условиях автономного вождения на городских дорогах в данном исследовании предлагается инновационный метод прогнозирования внимания, основанный на отслеживании движения глаз. Традиционное определение позы всего тела страдает от значительных ошибок прямой видимости (±15°) и плохо работает с небольшими, удалёнными целями, например размером глазного яблока всего 10×10 пикселей. В отличие от этого, данное решение использует стандартную RGB-камеру и обеспечивает эффективное восприятие благодаря двухрежимной архитектуре. В режиме RAW система напрямую использует Mediapipe FaceMesh для извлечения опорных точек лица и объединяет алгоритм PnP для оценки положения головы. Основным компонентом является усовершенствованная модель оценки взгляда EfficientNet-B0. Для улучшения извлечения периорбитальных признаков реализован облегчённый механизм внимания, а модуль симуляции расстояния предназначен для имитации ухудшения дальних визуальных признаков, что значительно повышает надёжность при низких разрешениях. Модель в режиме RAW обучается на наборе данных MPIIGaze. Экспериментальные результаты показывают, что угловая погрешность в реальном времени для целей на близком расстоянии (<5 м) составляет менее 0,1°, а алгоритм оценки взгляда позволяет эффективно различать ключевые состояния, такие как «взгляд прямо перед собой», «взгляд вбок» и «голова опущена». По сравнению с моделью лица YOLOv8s, обученной на архитектуре YOLOv8s и наборе данных WIDER FACE, предлагаемый метод обеспечивает повышение скорости примерно на 30 кадров в секунду при обнаружении одной цели и сохраняет преимущество в 3 кадра в секунду при обнаружении нескольких целей (≥5 человек). Этот подход представляет собой более перспективную альтернативу традиционным методам определения поз, предлагая недорогую и высокоэффективную поддержку безопасности в реальном времени для принятия решений автономным вождением в сложных городских условиях.

Ключевые слова: прогнозирование намерений пешеходов, автономное вождение в городских условиях, отслеживание движения глаз, оценка направления взгляда, низкая угловая погрешность, высокоскоростной вывод, недорогая RGB-камера

Research on pupil tracking techniques for autonomous driving systems

Z. Wang, D. D. Zhdanov, A. D. Zhdanov
ITMO University, Saint-Petersburg, Russian

Abstract. In response to the challenging problem of pedestrian intention prediction in urban road autonomous driving, this study innovatively proposes an attention prediction method based on eye tracking. Traditional full-body pose detection suffers from significant line-of-sight errors (±15°) and performs poorly for small, distant targets–for instance, the eyeball may be as small as 10×10 pixels. In contrast, this solution employs a standard RGB camera and achieves efficient perception through a dual-mode architecture. In RAW mode, it directly utilizes Mediapipe FaceMesh to extract facial fiducial points and combines the PnP algorithm to estimate head pose. The core component is an improved EfficientNet-B0 gaze estimation model. A lightweight attention mechanism is introduced to enhance periorbital feature extraction, and a distance-simulator module is designed to simulate the degradation of long-distance visual features, thereby significantly improving robustness at low resolutions. The RAW mode model is trained on the MPIIGaze dataset. Experimental results show that the real-time angular error for short-distance (<5m) targets is less than 0.1°, and the gaze estimation algorithm can effectively distinguish the key states such as "looking straight ahead", "side looking" and "head down". Compared with the YOLOv8s-face model trained on the YOLOv8s architecture and the WIDER FACE dataset, the proposed method achieves a speed improvement of approximately 30 FPS for single-target detection and maintains a 3 FPS advantage in multi-target detection scenarios (≥5 people). This approach represents a more forward-looking alternative to traditional pose detection methods, offering low-cost, high-real-time safety support for autonomous driving decisions in complex urban environments.

***Keywords*:** pedestrian intention prediction, urban autonomous driving, eye tracking, gaze estimation, low angular error, high-speed inferenc, low-cost RGB camera

Introduction

The analysis of human postures represents a key research direction within the current field of artificial intelligence. This area of study originally emerged from the development of robotics. In today's era of rapid artificial intelligence advancement, a fully functional artificial intelligence system should not be confined to server-based operations or voice interaction capabilities. Rather, it should embody a humanoid form and possess autonomous learning abilities. Consequently, human posture detection technology has garnered

increasing attention in recent years. A comprehensive understanding of human posture information is essential for its effective integration into intelligent robotic systems. Moreover, in-depth research on human behavior and postures enables artificial intelligence to more accurately interpret user intentions and even anticipate future human actions. Currently, human action recognition primarily relies on video data, as it offers continuous sequences of motion frames. Artificial intelligence systems detect and classify human subjects within these video sequences, analyzing changes between adjacent frames to identify specific actions. In contrast, posture prediction presents a greater challenge. It requires the system to accurately recognize current postures even when dealing with incomplete time-series data and to forecast subsequent action trends based on this recognition. Therefore, research on posture prediction holds significant practical value and has broad application potential, particularly in the areas of home safety for infants, children, and the elderly. For instance, posture recognition can be employed to issue early warnings for incidents such as falls. Furthermore, this technology significantly supports the development of autonomous driving systems by enhancing vehicles' ability to predict pedestrian behavior.

In existing research, it is already feasible to interpret the command intentions of traffic police by identifying their body postures and further predicting the lane-changing gestures they are about to perform [1]. This capability enables vehicles to avoid congested or under-construction road sections and prevents them from entering incorrect driving routes. Although current traffic police action recognition is primarily limited to a few predefined postures–typically identified by detecting key joint positions and matching them with pre-stored action templates in a database–it is evident that this technology holds significant potential for future development. At present, research and applications of autonomous driving systems are becoming increasingly mature. On highways, Level 3 assisted driving functionality has been largely realized, allowing drivers to allocate minimal attention to perform basic autonomous driving tasks. However, autonomous driving technology still encounters numerous challenges in urban road environments and has yet to achieve substantial breakthroughs. The primary reason lies in the abundance of unpredictable dynamic factors present in urban settings, such as pedestrians and cyclists, whose behaviors often lack regular patterns and are difficult to model accurately using traditional methods. Additionally, compared to highways, urban areas contain a high density of intersections, including alleys and entrances to residential zones. These locations are numerous, complex, and highly variable, making autonomous navigation riskier and more challenging. If posture detection technology can be leveraged to accurately predict the future actions of pedestrians, it would significantly enhance the adaptability of autonomous driving systems in urban environments, thereby enabling broader real-world applications.

Currently, autonomous vehicles generally employ a method based on the extraction of overall target features to detect pedestrians. By utilizing the significant differences in shape, size, and color between pedestrians and other traffic participants, semantic segmentation is performed to identify pedestrians [2]. However, this approach often results in a relatively large region of interest (ROI), which introduces a substantial amount of redundant data. This, in turn, increases computational load and reduces processing efficiency. Therefore, narrowing the ROI has become crucial for improving detection efficiency. This study proposes a novel approach that focuses on feature extraction from the head and facial regions of pedestrians. On one hand, since adults are typically taller than 1.2 meters and children are usually accompanied by adults, the area below 1.2 meters from the ground can be reasonably ignored, allowing detection efforts to be concentrated on regions above this height. The head region not only possesses distinct features that differentiate pedestrians from other objects on the road, making it easier to identify, but also provides valuable information. Specifically, the head point cloud data obtained through radar can offer an initial estimation of the pedestrian's orientation. This enables the system to infer whether the pedestrian is aware of the autonomous vehicle and whether sudden movement is likely, which could affect the vehicle's path planning. This method not only accelerates scene modeling during the perception stage but also enhances the accuracy and safety of the vehicle's path planning and risk prediction to a certain extent.

Statement of the problem

In previous studies, eye-tracking technology has been widely utilized as a critical method for monitoring the driver's state. By accurately capturing key ocular features, it provides quantitative data to support fatigue detection and attention evaluation. Currently, a prevalent approach employs a cascaded detection framework: first, a robust facial detection algorithm (e.g., YuNet) identifies the facial region; subsequently, the eye ROI is localized

within this area; finally, a gaze estimation model (e.g., the MPIIGaze training framework) generates a three-dimensional gaze direction vector. Compared to the conventional HOG (Histogram of Oriented Gradients) combined with SVM (Support Vector Machine) method, this architecture demonstrates significant improvements in real-time performance and adaptability to complex environmental conditions [3]. Particularly in challenging scenarios involving illumination variations or head pose deviations, it maintains reliable detection stability.

The research presented in this paper introduces eye tracking technology into the detection of pedestrians outside the vehicle. In urban traffic scenarios, the ability to assess pedestrians' attention states directly influences the decision-making safety of autonomous vehicles. The system faces three primary challenges. First, long-distance detection is critical for safety: when a vehicle is traveling at 60 km/h on urban roads, it requires a braking distance of 20–30 meters. Therefore, the detection range must be at least 30 meters to ensure sufficient response time. However, at such distances, the eye region may occupy only 10×10 pixels, rendering traditional eye keypoint detection ineffective. Second, the system must accurately analyze the direction of gaze, even when the head is turned. Third, it must determine whether the pedestrian is aware of the approaching vehicle. Conventional approaches rely on full-body pose estimation methods, such as OpenPose, but these techniques can result in gaze direction errors as high as ±15°. To address these challenges, this paper integrates the gaze estimation model MPIIGaze and transfers its robust gaze decoupling capability into the in-vehicle external perception system.

Object Detection in Autonomous Driving

The key to autonomous driving technology lies in enabling vehicles to accurately perceive their surrounding environment and make autonomous decisions. Within this system, the object detection algorithm serves as the core component of the perception module, directly influencing the vehicle's understanding of its surroundings and its ability to respond effectively. In recent years, with continuous advancements in deep learning and multi-modal sensing technologies, object detection methods in autonomous driving have evolved from initial single-image recognition approaches to integrated architectures that fuse data from multiple sensors. This evolution has enabled systems to efficiently identify and accurately locate objects such as vehicles, pedestrians, and obstacles in complex and dynamic traffic environments. However, challenges such as variations in lighting conditions, adverse weather, object occlusion, and the detection of small-sized objects remain significant technical hurdles. Currently, commonly used object detection methods in autonomous driving include the You Only Look Once (YOLO) algorithm [4], two-stage detection models (e.g., Fast R-CNN) [5], Transformer-based architectures [6], multi-sensor fusion models [7], and end-to-end (E2E) learning models [8].

The environmental perception capability of autonomous driving systems largely relies on the collaborative integration of multi-modal sensors and the input of high-quality data. Currently, mainstream autonomous driving vehicles are typically equipped with cameras, light detection and ranging (LiDAR) systems, millimeter-wave radars, and infrared sensors, forming a perception architecture that captures a wide range of physical characteristics. Specifically, cameras capture detailed texture and color information, LiDAR generates high-precision three-dimensional spatial point cloud data, millimeter-wave radars maintain stable performance under adverse weather conditions, and infrared sensors enhance visual perception in low-light environments. This multi-sensor fusion framework not only improves the system's adaptability to complex and dynamic environments but also enhances the redundancy of perceptual data. However, it also introduces significant challenges in the spatio-temporal alignment and fusion of heterogeneous data sources.

Table 1. **Characteristics of Common Sensors in Autonomous Driving Systems**

Type	Data	Advantageous scenarios	Main limitations	Sampling rate
Visible light camera	RGB array	Texture recognition, semantic understanding	Light-sensitive and weak night vision	30-60 FPS
LiDAR	3D point clouds	Precise distance measurement, 3D reconstruction	Rain and snow scattering, high cost	10-20 Hz
Millimeter-wave radar	Distance-Doppler matrix	Penetrate rain and fog, speed measurement	Low resolution, no height information	5-20 Hz
Infra-red sensor	Thermal radiation intensity map	Nighttime target, biological detection	Colorless texture, affected by heat sources	30-60 FPS

In terms of data representation, object detection algorithms must address the challenge of mapping between two-dimensional image space and three-dimensional real space. The point cloud data acquired by LiDAR is typically represented in the form of three-dimensional coordinates:

$$P = \{p_i \in \mathbb{R}^3 \mid i = 1, \dots, N\}.$$

Among these points, each point p_i is represented by its spatial coordinates (x, y, z), as well as the corresponding reflectance intensity I. The camera data is then projected onto the image plane based on the pinhole camera model:

$$s\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} [R \quad T] \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}.$$

The key to multimodal fusion is establishing an effective spatio-temporal alignment mechanism across different sensors. For instance, calibration enables the projection of LiDAR point clouds onto the image plane, generating depth-enhanced features.

1. Current Object Detection Algorithm

In recent years, the perception module of autonomous driving systems has undergone rapid development, evolving into a multi-paradigm framework that integrates visual, LiDAR, and multi-sensor fusion techniques. Current mainstream object perception algorithms can be broadly divided into four categories.

Vision-driven 2D/3D detection has become the most widely adopted approach due to the maturity of deep learning in computer vision. The YOLO series, as a representative, leverages end-to-end convolutional neural networks to achieve real-time inference with high accuracy. Its enhanced variants, such as YOLOv8n-CSS with attention mechanisms and multi-scale pooling, significantly improve the recognition of small objects in complex urban environments [9]. In parallel, stereo vision extends monocular detection by estimating object depth through binocular disparity maps, thereby enhancing detection performance under nighttime or low-light conditions [10].

Point cloud–based 3D object detection plays a central role in leveraging LiDAR data. Voxelization-based methods, exemplified by PointPillars, convert sparse point clouds into pseudo-images for efficient processing with 2D CNNs. Alternatively, point-based methods such as PointNet++ directly learn hierarchical local features from raw point sets, enabling more precise geometric representation. Despite their accuracy, these methods remain sensitive to weather-induced noise, necessitating domain adaptation and point cloud denoising strategies to maintain robustness [11].

Multi-sensor fusion frameworks are increasingly recognized as indispensable for achieving reliable perception in diverse real-world scenarios. By integrating complementary sensing modalities–cameras for texture, LiDAR for geometry, millimeter-wave radar for robustness in adverse weather, and infrared sensors for night vision–fusion strategies at the data, feature, or decision level provide redundancy and adaptability. Pre-fusion approaches, such as MV3D, combine depth maps with RGB features before convolutional processing, while post-fusion methods integrate independently processed detection outputs via optimization algorithms like Non-Maximum Suppression.

Bird's-Eye View (BEV) perception represents a novel paradigm that unifies heterogeneous sensor inputs into a top-down spatial representation [12]. BEV facilitates global consistency in target localization and scene understanding, thereby supporting advanced planning and decision-making modules [13]. Industrial implementations such as Tesla's HydraNet and XPeng's NGP 3.0 exemplify the state-of-the-art, combining image features, LiDAR point clouds, and high-definition maps to construct dynamically updated BEV grids with strong cross-view association.

In summary, the mainstream perception algorithms for autonomous driving are converging toward hybrid frameworks that balance the efficiency of visual methods, the precision of LiDAR-based 3D detection, and the robustness of multi-modal fusion. The integration of BEV-based representations further enhances spatial awareness, paving the way for end-to-end perception–planning pipelines in next-generation autonomous driving systems.

2. RAW Model

The RAW model begins with the acquisition of camera frames and proceeds through a series of processing stages, including multi-person facial key point and iris point extraction, eye-axis-oriented cropping and

normalization, a lightweight batch forward pass through the gaze estimation network, direction determination via the fusion of geometric and learning-based signals, stabilization using hysteresis and multi-frame consistency, and finally, visualization via discretized small arrows and log recording–thereby forming a fully end-to-end real-time system. First, MediaPipe FaceMesh (with refine_landmarks enabled) is employed to extract 468 facial landmarks and four high-precision iris coordinates per eye for each frame. For any detected face, let the pixel coordinates of the outer and inner corners of the right eye be denoted as P_{out}^{r} and P_{in}^{r}, respectively, and those of the inner and outer corners of the left eye as P_{in}^{l} and P_{out}^{l}, respectively. Based on these points, the "horizontal axis" vectors for both eyes are then constructed and normalized:

$$u_r = \frac{P_{in}^{r} - P_{out}^{r}}{\left\| P_{in}^{r} - P_{out}^{r} \right\|}, \qquad u_l = \frac{P_{out}^{l} - P_{in}^{l}}{\left\| P_{out}^{l} - P_{in}^{l} \right\|}.$$

To ensure that the subsequent network processes eyes with a consistent orientation, we construct an affine transformation matrix M at the midpoint $c = \frac{1}{2}(P_1 + P_2)$ between the eyes, based on the rotation angle $\theta = \tan^{-1}2(\Delta y, \Delta x)$ of the line connecting the eye corners. An "orientation-aware cropping" is then applied to a square local window using the function warpAffine(·;M). The window's side length is determined proportionally to the inter-pupillary distance and constrained within a reasonable range. After aligning both eye regions, each local image is converted to grayscale, resized to 224×224, and linearly normalized using $(x - 0.5)/0.5$. These images are then stacked across all samples and both eyes into a batch tensor of shape $[N, 1, 224, 224]$. We adopt the `channels_last` memory format and utilize CUDA's `autocast` for half-precision computation. A single forward pass is executed under `torch.inference_mode()` to minimize CPU/GPU data transfer overhead and scheduling latency. This preprocessing pipeline ensures that the network consistently receives eye patches with aligned geometric poses and uniform scales, thereby significantly reducing domain shift caused by minor variations in head pose.

The gaze regression network adopts EfficientNet-B0 as the backbone, with the first convolutional layer modified to accept single-channel grayscale input. At a mid-level feature layer with 24 channels in the backbone, a 1×1 gating attention mechanism is introduced. This mechanism first compresses the channel dimension using a 1×1 convolution followed by a ReLU activation, and then applies another 1×1 convolution followed by a Sigmoid activation to generate channel-wise weights. These weights are then applied via element-wise multiplication with the original feature maps, effectively emphasizing channels that contain stronger eye anatomical cues. Following this, a 1×1 "distance simulation" convolution is employed to reduce the feature dimensionality to 256 channels. After applying adaptive pooling to resize the features to 4×4 and flattening the output, two fully connected layers are used to regress the 3D gaze vector. Finally, L2 normalization is applied to the output to ensure a consistent vector magnitude, representing a directional signal. To align the "looking straight ahead" direction across individuals in the vector space, the system implements a "center calibration" procedure. Upon triggering the calibration, the system computes the average gaze vector from a short sequence of recent frames. This vector is then rotated to align with the forward direction of the camera using Rodrigues' rotation formula. The resulting vector is transformed into screen coordinates, where the x-axis increases to the right and the y-axis increases upward. The resulting offset is stored as the individual's zero-point reference. For each subsequent frame, the network output is first aligned to this reference before computing the final gaze angle and direction.

The vertical direction is entirely determined by the pitch angle derived from the network. Let the aligned unit vector be denoted as $s = (s_x, s_y, s_z)$, and the pitch angle is represented by

$$pitch = \tan^{-1}2(s_y, s_z) \cdot \frac{180}{\pi}.$$

Subtract $pitch_0$ obtained during center calibration from the current measurement to obtain $\Delta pitch$. To suppress critical jitter, the system defines distinct entry and exit thresholds for the Up and Down directions, thereby creating a hysteresis loop. A certain direction is entered only when the entry threshold is reached, and to return to a non-target state, the signal must fall back below the tighter exit threshold. This mechanism effectively prevents rapid oscillations such as "up/center/up/center" caused by micro-vibrations and environmental noise in real-world conditions.

For the left - right direction, instead of directly using the yaw angle of the network, the geometric "iris offset difference method" is adopted. Inside each eye, taking the line connecting the corners of the eye as

the axis, the centroid of four iris points is calculated and projected onto this axis. For the right eye, the positive direction is from the outer corner to the inner corner, and for the left eye, the positive direction is from the inner corner to the outer corner. Thus, two dimensionless offset quantities "relative to the mid - point of the eye corners" are obtained, which respectively describe the "inward" and "outward" displacements. In order to symmetrically fuse the information of both eyes, the system defines the horizontal difference scalar:

$$d_h = \frac{1}{2}(c_r - c_l).$$

Among them, c_r denotes the offset of the right eye with “inward as positive”, and c_l denotes the offset of the left eye with “outward as positive”. Therefore, $d_h > 0$ indicates a rightward gaze, while $d_h < 0$ indicates a leftward gaze. Considering the potential interference caused by pixel noise and eyelid occlusion, dh is not used directly. Instead, it is passed through a first-order IIR low-pass filter to achieve signal smoothing. The filter is updated only when the eye aspect ratio (EAR) exceeds a predefined threshold, which indicates that the eyes are fully open. The filtering process follows the equation:

$$\widetilde{d_h}[k] = (1 - \alpha)\widetilde{d_h}[k-1] + \alpha d_h[k].$$

Among them, α denotes the smoothing coefficient. The EAR (Eye Aspect Ratio) measures the degree of eye opening and closing by calculating the ratio of "the average vertical distance between the upper and lower eyelids to the horizontal distance between the outer eye corners". This approach helps prevent misjudgment in cases where facial keypoints are detected but the iris is temporarily occluded. The system defines the "direct gaze central zone" as the region where $|\Delta pitch|$ is within a predefined pitch bandwidth and $|\widetilde{d_h}|$ is within a specified horizontal bandwidth. If the gaze falls within this central zone, the current frame's $\widetilde{d_h}$ value is added to the "direct gaze sample pool". The standard deviation σ of this sample pool is then used to determine an adaptive threshold: entry thresholds for left and right directions are set based on $K\sigma$, while the exit threshold is defined as a fixed proportion of the entry threshold. This mechanism enables automatic adjustment of sensitivity to match individual noise levels, balancing responsiveness ("easy to trigger") with stability ("not overly sensitive").

When the up-down and left-right channels each provide candidate directions, the system does not simply follow the principle of "whoever triggers first prevails". Instead, it calculates the intensity of each relative to its respective entry threshold and selects the stronger side as the "principal axis". When the two sides are similar, the system preferentially retains the principal axis from the previous frame to reduce the back-and-forth flickering of the "diagonal state". A candidate direction must remain consistent across several consecutive frames before it is confirmed as the final label. This multi-frame confirmation mechanism is particularly effective in mitigating false transitions caused by glare, slight head movements, or brief keypoint jitter. Considering that most front-facing cameras display images in a mirrored manner, the system applies mirroring compensation at the left-right label level to align the on-screen arrow direction with the user's visual intuition. During the visualization stage, the continuous three-dimensional gaze vector is no longer used to render the "tracking arrow", as this would amplify and display the system's internal noise through arrow oscillation, potentially causing misleading visual illusions. The current approach directly maps the final labels into small discrete arrows on the screen plane: Left is represented by a short, stable left-pointing arrow, Right by a right-pointing arrow, Up by an upward arrow, and Down by a downward arrow. The arrow length is adjusted slightly based on the "intensity," albeit within a very narrow range. For the Direct mode, no arrow is displayed; instead, only the eye key points and a minimalist HUD are shown. This maintains the intuitive perception of "appearing to point at a location" while minimizing psychological discomfort caused by visual jitter. In the upper-left corner of the interface, only the current model mode, the two-second sliding window average FPS, and the number of detected faces are displayed. All other intermediate variables are logged into a CSV file, facilitating subsequent comparisons of robustness, angular error, and efficiency across different parameter configurations.

To maintain time consistency in a multi-person scenario, the system maintains a set of lightweight trajectories. Each trajectory uses the "eye center" from the previous frame as an anchor point. In the new frame, this anchor point is matched to the newly detected "eye center set" using the nearest neighbor method. If the matching distance exceeds a predefined threshold, a new trajectory is initiated. For existing trajectories that fail to find a match, the "missing count" is incremented, and if it exceeds a specified upper limit, the trajectory is discarded. Each trajectory independently stores the zero offset, threshold, adaptive statistics, and internal

filter state for the corresponding individual. As a result, even when multiple individuals alternate in and out of the frame, the calibration and threshold parameters remain free from mutual interference or "contamination." Additionally, the code includes a "demo mode" that utilizes ONNX face detection based on YOLOv8s, which runs on the GPU when the CUDAExecutionProvider is available. However, the main processing pipeline continues to focus on the iris and eyelid geometry provided by FaceMesh, as these are essential for the "iris offset difference method". In terms of performance, the main bottlenecks occur in FaceMesh on the CPU side and in image preprocessing. To mitigate this, we batch all individuals' eye patches and feed them into the model collectively. On the GPU side, we enable channels_last memory layout, half-precision computation, and cuDNN algorithm optimization, while limiting the number of OpenCV threads to allocate computational resources more efficiently toward forward passes. In the background, the system continuously accumulates per-frame processing times to estimate FPS. It records various metrics–including average FPS, angular error between the alignment vector and the forward axis, a proxy for robustness and false detection rate, directional binary accuracy, and the ratio of the minimum eye area to the total frame area–into a CSV file. Combined with several interactive components such as center calibration, mirror compensation, and mode switching, the entire pipeline is capable of not only sensitively responding to changes in gaze direction but also stably outputting discrete, application-friendly semantic outputs such as "Up/Down/Left/Right " in real, noisy, and multi-person front-camera environments. In conclusion, the RAW model offers several advantages over the currently prevalent object detection models.

Table 2. **The main innovations of the RAW model**

Model design	RAW	Others
Input and Key Points	Directly employed MediaPipe FaceMesh, which consists of 468 facial landmarks and 4 iris landmarks, for real-time processing of the video stream.	Make more use of pre-annotated datasets; key points/irises are often used in offline annotation or weak supervision.
Eye pretreatment	"Crop along the eye axis", scaled equally to 224×224 grayscale patches.	Fixed-frame cropping or normalization to the eye/head coordinate system.
Model and Input	EfficientNet-B0 single-channel + 1×1 gated attention + lightweight regression head; output unitized 3D vector.	Multi-channel inputs such as ResNet/EfficientNet/Hourglass; sometimes combined with head pose input.
Adaptive Thresholding	d_h standard deviation was counted in the "direct-looking center band", and the left and right thresholds were adapted to the center bandwidth.	Relying on fixed thresholds or data-driven confidence thresholds.
Anti-blinking / Anti-obscuring	The EAR threshold was used to suppress d_h update. Keep the last stable state.	Time-domain filtering or robust loss.

Experimental results

During the evaluation phase, the system is implemented in two modes: the RAW mode and the YOLO mode. In the RAW mode, FaceMesh is directly applied to the full image to estimate the gaze vector. In contrast, the YOLO mode utilizes YOLOv8s-face to detect the pedestrian’s facial region before gaze estimation. From the perspective of autonomous driving applications, this pupil tracking approach is anticipated to deliver substantial performance enhancements. First, in terms of computational efficiency, focusing on the eye region significantly reduces the size of the ROI, thereby lowering the computational overhead associated with feature extraction. On the experimental platform based on the NVIDIA RTX 4070 Ti chip, the system achieves stable processing speeds of over 60 FPS for single-target detection, which is critical for real-time perception in in-vehicle systems. According to NVIDIA’s architectural documentation, the INT8 throughput of the Tensor Core is approximately eight times that of FP32. Based on this specification, the equivalent computing power of the NVIDIA RTX 4070 Ti can be estimated at approximately 360 TOPS.

Table 3. **Comparison of chip computing power**

Chip	INT8 TOPS	Power (W)	TOPS/W
RTX 4070 Ti	360	285	1.26
NVIDIA Orin	254	60	4.23
Tesla FSD (HW3)	144	72	2.00
Mobileye EyeQ Ultra	176	100	1.76

After conversion based on computing power ratio, taking the widely adopted NVIDIA Orin chip as an example, the single target detection speed of this algorithm when deployed on an automotive-grade chip is also greater than 40 FPS.

The RAW model demonstrated clear advantages in speed and stability when handling a single target. In RAW mode, the average frame rate reached 64 FPS, as shown in Figure 1. Since the maximum frame rate of the camera used in the experiment is limited to 30 FPS, the 64 FPS value listed here represents the frame rate achieved without accounting for camera blockage.

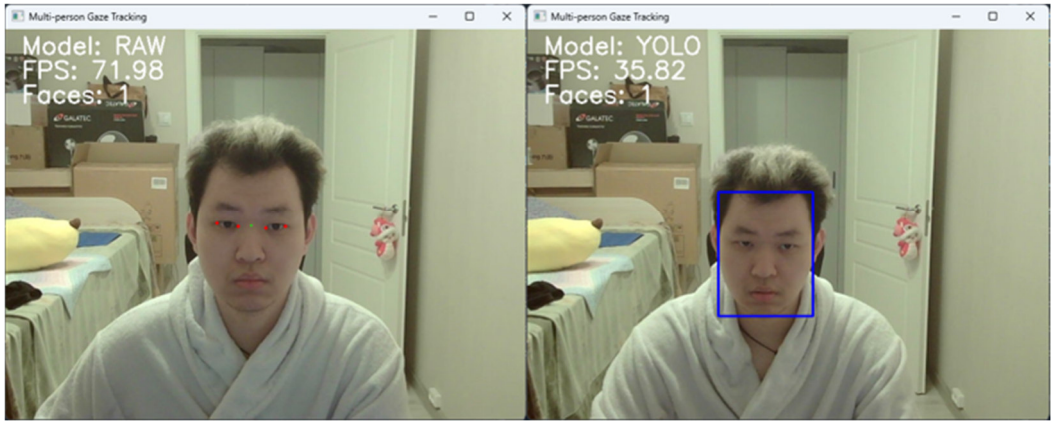

Figure 1. RAW modle and YOLOv8s-face

In the YOLO model family, the YOLOv8s-face model trained on the WIDER FACE dataset demonstrates the best performance. To further enhance the model's robustness during training, we incorporated data augmentation techniques such as random rotation, motion blur, and adjustments to brightness and contrast. As a result, the model achieved an average frame rate of 32 FPS in the single-object detection experiment. As illustrated in Figure 2, the frame rate comparison chart visually highlights the substantial differences in real-time performance between the two approaches.

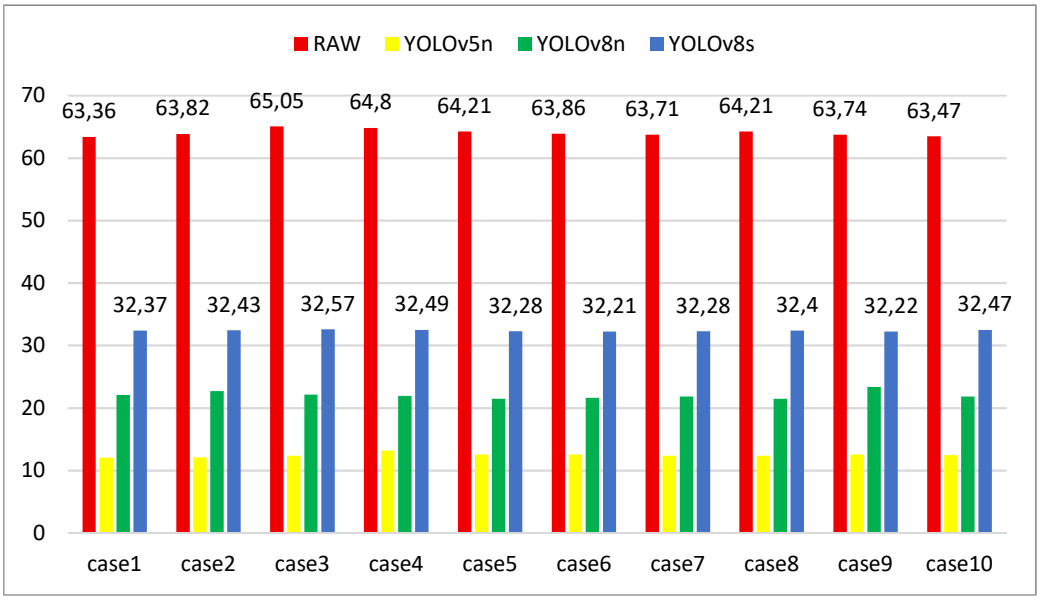

Figure 2. Comparison with other models (single object)

This difference suggests that in single-person or low-density population scenarios, pure eye tracking can substantially reduce computational demands while still fulfilling the real-time processing requirements of autonomous driving systems. Conversely, in environments involving multiple individuals or complex backgrounds, the performance of the RAW mode degrades more significantly due to differences in computational methodology. In multi-target real-time detection scenarios, the RAW mode relies on Mediapipe FaceMesh, which operates through a two-stage process: first, the lightweight BlazeFace model detects and localizes facial regions across the entire frame. Subsequently, for each detected target, 468 facial keypoints are independently regressed, followed by cropping of the eye region, which is then fed into the gaze estimation network. In single-person cases, this procedure exhibits nearly constant time complexity. However, as the number of targets increases, the operations involved in keypoint regression and ROI cropping accumulate linearly, leading to an overall time complexity that grows proportionally with the number of detected individuals:

$$T_{RAW}(N) = T_{det} + N \cdot (T_{mesh} + T_{gaze}).$$

Among them, T_{det} represents the global face detection time consumption, while T_{mesh} and T_{gaze} denote the time required for single-object keypoint regression and gaze prediction, respectively. When N is relatively large, the linear term becomes dominant, leading to a significant decrease in frame rate. Additionally, FaceMesh incorporates an internal temporal smoothing mechanism to stabilize the keypoint trajectory, utilizing a forward rolling window to fuse features across multiple frames. Consequently, during sudden increases in frame volume or in the presence of multiple objects, cache read/write operations and weighted updates further contribute to the computational load and introduce additional latency.

In contrast, the YOLO model employs a single-stage dense prediction architecture, in which the entire detection process involves only a single global feature extraction and multi-scale prediction. Consequently, its time complexity is approximately constant and can be roughly expressed as

$$T_{YOLO}(N) \approx T_{backbone} + T_{neck} + T_{head} + T_{NMS}.$$

Among them, the target number N introduces only a slight linear overhead during the NMS stage, which is typically much lower than the dominant computational cost associated with the feature extraction phase. Consequently, even when detecting dozens of targets in dense scenes, the frame rate of YOLO remains largely stable.

Discussion of results

The experimental results validate the aforementioned analysis. In a single-target scenario, utilizing frame_time to measure the data prior to imshow/waitKey (without accounting for potential camera blockage), the frame rate in RAW mode can achieve approximately 60 FPS. The angular error remains stable at around 0.05°, and due to temporal smoothing, the key-point stability is relatively high. The algorithm effectively controls the error range through the fusion and normalization of binocular gaze vectors. This enables reliable discrimination of key attention states such as "looking ahead", "looking sideways", or "looking down", as shown in Figure 3. Both RAW and YOLO modes are extremely robust in the single-player case, with no keypoint loss.

Figure 3. Key gaze direction recognition

In contrast, in the multi-target scenario (≥5 people), the frame rate of the RAW mode decreases to approximately 23 FPS, primarily due to the increased computational load from per-target key-point regression and the rolling window mechanism, but an accurate determination of the line-of-sight direction is still achieved, as illustrated in Figure 4. In multi-object detection, the YOLO mode does not introduce keypoint detection to obtain robustness of 1 for the number of detected faces/actual number of faces. In RAW mode, Boolean indications are adopted to assess whether the system is working properly. If FaceMesh has detected at least one face in a frame, it will be 1, otherwise it will be 0, some keypoint information will be lost when the angle is large. But the combined availability coverage coefficient will be no less than 0.98.

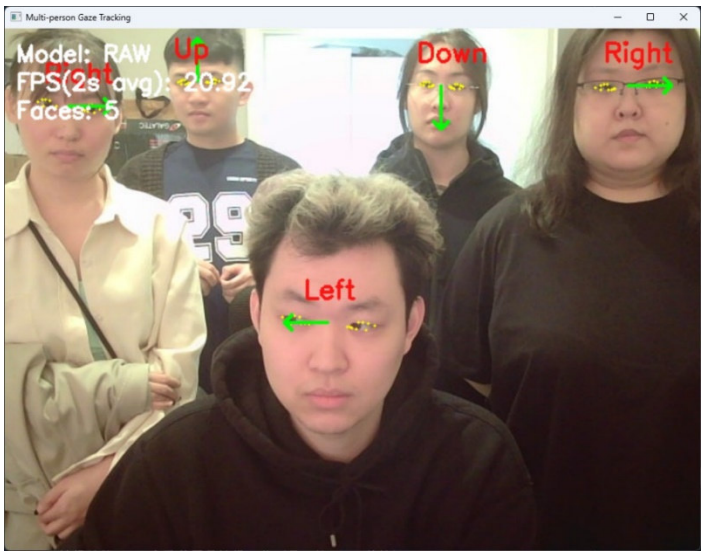

Figure 4. Gaze direction in multi-object scenariosGaze direction recognition in multi-object scenarios

Under the same conditions, the YOLO mode (YOLOv8s-face) exhibits a frame rate that remains relatively stable at around 30 FPS. The reduction from the original 32 FPS is minimal, and the accuracy in detecting the number of people remains largely unchanged, as illustrated in Figure 5.

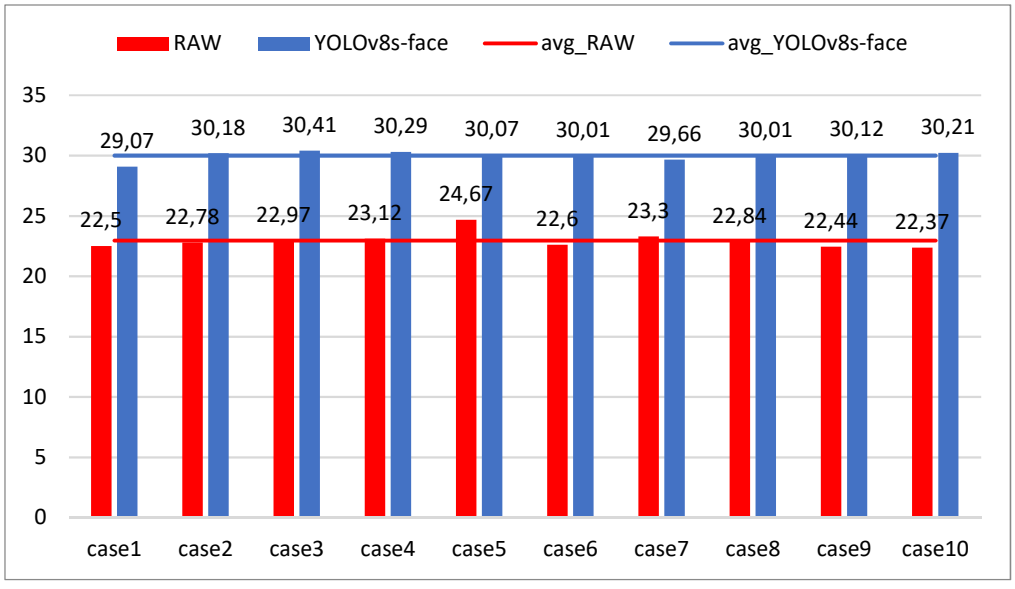

Figure 5. Comparison with YOLOv8s-face (multi-object)

Although the performance of the RAW mode degrades significantly in multi-target detection, it still outperforms the YOLO mode by maintaining a frame rate that is approximately 2 FPS higher. This clearly demonstrates the objective advantages of the RAW mode over the YOLO mode. Moreover, compared with YOLO, which only detects the faces of pedestrians for early warning, the RAW mode calculates the line of sight direction based on full-face recognition. As a direct manifestation of a pedestrian's attention, the line of sight

direction can predict potential intentions 2 to 3 seconds before a jaywalking behavior occurs. This provides a greater lead time than methods relying solely on human key points or movement trajectories, thereby enhancing risk prediction capabilities. In addition, in multi-target scenarios, the system can aggregate the gazing directions of multiple pedestrians to generate an "attention heat map," which assists the decision-making module in identifying the focal points of crowd attention. This feature is particularly valuable in complex environments such as traffic light intersections. More importantly, this method operates using only standard RGB cameras, without requiring additional infrared light sources or expensive dedicated hardware. It offers low cost and high scalability, making it suitable for large-scale deployment across various autonomous driving platforms, including passenger cars and urban buses.

In terms of accuracy, the average angular error of the model on the MPIIGaze validation set is approximately 13° to 14°, which is comparable to that reported in public studies such as Gaze360 and ETH-XGaze [14], [15]. In actual experimental settings, under ideal conditions including sufficient distance, good lighting, and a simple surrounding environment, the average angular error was found to be extremely small–only about 0.05°. However, even in less controlled, real-world scenarios such as vehicle-mounted applications. Such capability is crucial for autonomous vehicles to anticipate pedestrian intentions and make timely decisions regarding deceleration or avoidance.

Meanwhile, the issue mentioned previously–that the pedestrian's eye area may be smaller than 10×10 pixels at long distances–was also confirmed in the experiment. As shown in Figure 6, a standard 1080P computer camera was used, under which the eye region in the image was already significantly small. Specifically, the recorded eye area measured only 22×1 pixels when the subject was 3 meters away from the camera. Although the current performance is far from achieving the 30-meter recognition distance mentioned at the beginning of the paper, this study still offers novel insights into vehicle and pedestrian target detection systems. Future work should include experiments using high-resolution cameras and further algorithmic improvements to continuously optimize the model, with the ultimate goal of attaining performance that meets the requirements of in-vehicle systems.

Figure 6. The size of the eye area at the maximum distance-3 meters

It is worth noting that the mPA (mean Pixel Accuracy/mean Point Accuracy) metric, commonly utilized in facial keypoint detection or segmentation tasks, is not included in the primary comparison. This exclusion is due to the fact that the central objective of this study is 3D gaze direction prediction. Furthermore, the keypoint detection component employs the pre-trained Mediapipe FaceMesh model without any retraining or fine-tuning, and the accuracy of its detected keypoints has been rigorously validated in the original paper [16]. Additionally, there is no strong linear relationship between keypoint localization accuracy and the final gaze angle error. Hence, incorporating mPA into the evaluation would not provide an accurate reflection of the system's actual performance in gaze estimation. Consequently, this paper emphasizes the angle error as a direct indicator of gaze prediction quality, and complements it with metrics such as frame rate and robustness to offer a comprehensive assessment of the system's practicality and stability. Future work may consider integrating metrics like mPA or NME (Normalized Mean Error) to enable a more detailed analysis of error sources within the system.

Summary

In response to the challenges of pedestrian intention prediction in urban road autonomous driving, this study innovatively proposes an attention prediction method based on pupil tracking. An efficient perception framework is achieved through a dual-mode processing architecture: in RAW mode, Mediapipe FaceMesh is directly utilized to extract full-image facial key points and estimate head pose; in YOLO mode, YOLOv8s-face is employed to pre-detect pedestrian facial regions. The core of this method is an improved EfficientNet-B0 gaze estimation model. A lightweight attention mechanism is integrated to enhance feature responses around the eyes, while a distance-simulator module is introduced to simulate the degradation of long-distance visual features, thereby significantly improving robustness under low-resolution conditions. Experimental results demonstrate that in short-distance scenarios (<5 meters), RAW mode achieves an ultra-low angular error of 0.05° and operates at a real-time frame rate of 60 FPS for a single target, effectively distinguishing critical states such as "looking straight," "looking sideways," and "bowing the head." In multi-target scenarios involving five or more pedestrians, the system still maintains a frame rate of 30 FPS, which is 10 FPS faster than YOLO mode. The proposed method requires only a standard RGB camera, eliminating the need for infrared equipment. When deployed on the NVIDIA Orin automotive chip, the system achieves a processing speed of over 40 FPS and enables the prediction of pedestrian intentions to cross the road 200–300 milliseconds in advance, offering a low-cost, high-real-time solution for autonomous driving decision-making in complex urban environments. Future work will focus on improving detection performance for small targets (e.g., eye regions smaller than 10×10 pixels) at long distances (>30 meters) and exploring synergistic enhancements between high-resolution cameras and algorithmic improvements.

Reference

1. Ma, Nan, et al. "A survey of human action recognition and posture prediction." *Tsinghua Science and Technology* 27.6 (2022): 973-1001.

2. Zhang, Bing, et al. "Using artificial neural networks for human body posture prediction." *International Journal of Industrial Ergonomics* 40.4 (2010): 414-424.

3. Li, Yunyang, et al. "Eye-gaze tracking system by haar cascade classifier." *2016 IEEE 11th Conference on Industrial Electronics and Applications (ICIEA)*. IEEE, 2016.

4. Cheng, Siqiang, Lingshan Chen, and Kun Yang. "DGSS-YOLOv8s: A Real-Time Model for Small and Complex Object Detection in Autonomous Vehicles." *Algorithms* 18.6 (2025): 358.

5. Ren, Shaoqing, et al. "Faster R-CNN: Towards real-time object detection with region proposal networks." *IEEE transactions on pattern analysis and machine intelligence* 39.6 (2016): 1137-1149.

6. Beal, Josh, et al. "Toward transformer-based object detection." *arXiv preprint arXiv:2012.09958* (2020).7. Senel, Numan, et al. "Multi-sensor data fusion for real-time multi-object tracking." *Processes* 11.2 (2023): 501.

7. Wang, Ao, et al. "Yolov10: Real-time end-to-end object detection." *Advances in Neural Information Processing Systems* 37 (2024): 107984-108011.

8. Behley, Jens, et al. "Semantickitti: A dataset for semantic scene understanding of lidar sequences." *Proceedings of the IEEE/CVF international conference on computer vision*. 2019.

9. Roy, Payel, et al. "Adaptive thresholding: A comparative study." *2014 International conference on control, Instrumentation, communication and Computational Technologies (ICCICCT)*. IEEE, 2014.

10. Lang, Alex H., et al. "Pointpillars: Fast encoders for object detection from point clouds." *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019.

11. Li, Hongyang, et al. "Delving into the devils of bird's-eye-view perception: A review, evaluation and recipe." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 46.4 (2023): 2151-2170.

12. Zhou, Hao, et al. "Review of learning-based longitudinal motion planning for autonomous vehicles: research gaps between self-driving and traffic congestion." *Transportation research record* 2676.1 (2022): 324-341.

13. Zhang, Xucong, et al. "Appearance-based gaze estimation in the wild." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015.

14. Kellnhofer, Petr, et al. "Gaze360: Physically unconstrained gaze estimation in the wild." *Proceedings of the IEEE/CVF international conference on computer vision*. 2019.

15. Zhang, Xucong, et al. "Eth-xgaze: A large scale dataset for gaze estimation under extreme head pose and gaze variation." *European conference on computer vision*. Cham: Springer International Publishing, 2020.

16. Zhang, Xucong, et al. "Mpiigaze: Real-world dataset and deep appearance-based gaze estimation." *IEEE transactions on pattern analysis and machine intelligence* 41.1 (2017): 162-175.