Using YOLOv7 Combined with Visual Attention Mechanism to Achieve Real-time Tracking of Passenger Gaze Targets in Urban Rail Interactive Guidance Interface

Leilei Zhai[1,2], Xuehong Wan[2], Yimeng Zhao[2], Lifeng Li[1], Liuxi Wu[3]
[1]School of Traffic and Transportation, Beijing Jiaotong University, Beijing 100044, China
[2]Beijing Urban Construction Design & Development Group Co., Limited, Beijing 100037, China
[3]Institute of Urban and Sustainable Development, City University of Macau, Macau 999078, China
Corresponding author: Leilei Zhai (e-mail: lleizhai@163.com)

ABSTRACT Real-time gaze target tracking in urban rail interactive guidance interfaces faces significant challenges due to multitarget occlusion, limited computational resources, and the need for low-latency visual perception. This study proposes a lightweight tracking framework that integrates an improved YOLOv7 detector with the Convolutional Block Attention Module (CBAM) and an enhanced ByteTrack algorithm to achieve efficient and accurate passenger gaze target localization. MobileNetV3 is adopted as the backbone network to reduce computational complexity, while BiFPN and Gaussian Soft-NMS improve multi-scale feature fusion and detection robustness. MediaPipe and PnP-based gaze estimation are combined with an NSA Kalman filter and GIoU matching strategy to enhance temporal consistency and tracking stability under complex scenarios. Experimental results demonstrate that the proposed framework achieves a mAP@0.5 of 96.3%, a MOTP of 85.7%, and 32.2 FPS on embedded platforms, maintaining reliable performance under severe occlusion conditions. Beyond intelligent transportation applications, the proposed lightweight visual perception and attention modeling framework provides useful methodological references for adaptive sensing, edge intelligence, and real-time signal interpretation in electromagnetic wave propagation environments and antenna-assisted human– machine interaction systems.

INDEX TERMS lightweight object detection, gaze target tracking, visual attention mechanism, electromagnetic sensing, edge intelligence

I. INTRODUCTION

WITH the rapid development of urban rail transit, the interactive guidance interface in station halls and carriages has become an important window for passengers to obtain information; similarly, in the industry, advanced digital interfaces and monitoring displays have become a critical window for operators to obtain real-time data on production line status, material flow, and quality control metrics [1]. However, during peak hours, with dense crowds, complex backgrounds, and multiple screens, the system struggles to accurately locate the interface elements passengers are looking at. This results in delayed responses or false touches on the urban rail interactive guidance interface, which in turn affects passengers' user experience and travel efficiency. Ensuring high-precision tracking of the target while meeting the real-time requirements of the system on embedded or edge devices in dynamic multi-target occlusion scenarios has become a key technical problem that needs to be solved urgently.

This research aims to develop a technical solution for accurate, real-time tracking of passenger gaze targets in urban rail interactive signage interfaces. This approach replaces the YOLOv7 (You Only Look Once version

7) backbone with a lightweight MobileNetV3. It integrates a Convolutional Block Attention Module (CBAM) into the neck to improve feature response for small objects and occluded areas. Then, using BiFPN (Bidirectional Feature Pyramid Network) multi-scale fusion and Soft-NMS (Soft Non-Maximum Suppression) decoding optimization to further enhance detection robustness. The passenger's line of sight is then mapped to the guidance interface through the PnP (Perspective-n-Points) posture solution algorithm to complete the positioning and tracking of the gaze target. Finally, based on the improved ByteTrack, the NSA (Neural Networkbased Appearance) Kalman filter and the GIoU (Generalized Intersection over Union) matching strategy

are introduced to achieve adaptive prediction and accurate association of motion states.

II. RELATED WORK

Many scholars have conducted research in the field of urban rail passenger detection and tracking. Li X et al. introduced bidirectional feature fusion and adaptive spatial feature fusion into YOLOv4, which improved mAP from 89.63% to 92.96% and reduced detection time from 61ms to 52ms, but the model was still relatively heavy [2]. Junyan W U et al. combined YOLOv5 with DeepSORT (Deep Simple Online and Real-time Tracking) to extract head and shoulder features, achieving 86% accuracy and 35FPS, but still missing detections under occlusion [3]. Zhang W et al. integrated M DeepSORT (Modified-Deep Simple Online and Real-time Tracking) with YOLOv5. They improved stability through Kalman filtering and momentum trajectory update, but the process was complicated [4]. Sakthi S M implemented car density statistics in videos based on YOLO, but did not optimize accuracy and occlusion processing [5]. Li N et al. proposed GR YOLO (Giou-RepC3 -You Only Look Once version), which improved mAP by 3–11%, but the number of parameters was still large [6]. Wang H improved YOLOv4 by introducing region of interest enhancement, achieving 93.52% mAP and 25FPS, but insufficient support for multi-target interaction scenarios [7]. Zuo J used binocular vision and YOLO to locate the standing position of passengers, with an error of 18.1cm and a time consumption of 860ms, and weak real-time performance [8]. Zidani G achieved an MOTA (Multiple Object Tracking Accuracy) of over 63 and an HOTA (Higher Order Tracking Accuracy) of over 41 under the YOLOv8 and DeepSORT frameworks, with enhanced occlusion handling, but it has not been verified in track scenarios [9]. Existing research has made some improvements in accuracy and real-time performance, but it relies heavily on heavy models and multistage processes, making it difficult to meet the lightweight and robustness requirements under resource constraints and complex occlusion conditions. A more efficient and integrated solution is urgently needed. In terms of passenger gaze target tracking, Li L et al. explored the influence of environmental factors on gaze behavior and boarding intention in waiting scenes through eye movement experiments, but the research was limited to static images [10]. Zeng Z et al. combined eye tracking and spatial syntax to analyze the impact of high-speed rail station guide signs on passengers' wayfinding behavior, but did not involve dynamic tracking technology [11]. Kim JK et al. proposed symmetrical angle magnification and central gravity function to improve the accuracy of edge gaze on large screens, but this method relies on personalized calibration and is difficult to adapt to real-time lightweight systems [12]. Existing research has mostly focused on psychological behavior and static visual analysis, and lacks technical support for real-time tracking of gaze targets suitable for dynamic interactive interfaces.

III. LIGHTWEIGHT YOLOV7 DETECTION AND PASSENGER GAZE TARGET MAPPING

A. ORIGINAL YOLOV7 ALGORITHM

YOLOv7 is a single-stage object detection framework that combines efficient feature extraction, feature fusion, and multi-scale prediction mechanisms [13,14]. In the feature extraction stage, the core of YOLOv7′s unique enhanced ELAN (Efficient Layer Aggregation Network) structure is a stacked, scalable residual block. The feature extraction stage formula is shown in (1).

$$F_{i+1} = \sigma\left(\mathrm{BN}(C(F_i))\right), \quad i = 0, 1, \ldots, N-1 \tag{1}$$

In formula (1), C represents the convolutional unit; BN represents batch normalization; and F_{i+1} represents the feature map output by feature extraction.

YOLOv7 uses the spatial pyramid pooling fusion module SPPF (Spatial Pyramid Pooling Fast) to apply to the feature map and generate multi-receptive field features, as shown in formula (2).

$$F_{\mathrm{SPPF}} = \mathrm{Concat}\left(F_N, P_5(F_N), P_9(F_N), P_{13}(F_N)\right) \tag{2}$$

In formula (2), P represents the maximum pooling operation, and F_{SPPF} represents the multi-receptive field feature.

During the feature fusion stage, a top-down Path Aggregation Network (PANet) is used to achieve cross-scale information transfer [15,16]. The bottom-up and top-down paths are represented as shown in formula (3).

$$\begin{aligned} F_l^{\mathrm{up}} &= \mathrm{Upsample}(F_{l+1}) + F_l, \\ F_l^{\mathrm{down}} &= \mathrm{Downsample}(F_{l-1}) + F_l. \end{aligned} \tag{3}$$

In formula (3), F_l represents the feature map of the l-th layer; Upsample represents upsampling through bilinear interpolation; and Downsample represents downsampling through convolution with a stride of 2. During the prediction phase, YOLOv7 performs anchor box predictions in parallel on feature maps at three scales (80, 40, and 20). Each scale feature map includes an anchor box, and each anchor box outputs five positioning parameters (bounding box center relative to the cell offset, bounding box width and height logarithmic scale transformation, object confidence, and category confidence.

The actual position conversion formula of the bounding box is shown in formula (4):

$$\begin{aligned} b_x &= \sigma(\hat{t}_x) + \alpha_x, \quad & b_y &= \sigma(\hat{t}_y) + \alpha_y, \\ b_w &= p_w e^{\hat{t}_w}, & b_h &= p_h e^{\hat{t}_h}. \end{aligned} \tag{4}$$

In formula (4), $\hat{t}_x$ and $\hat{t}_y$ represent the offset of the bounding box center relative to the cell, $\hat{t}_w$ and $\hat{t}_h$ represent the logarithmic scale transformation of the bounding box width and height. α_x and α_y represent the coordinates of the upper left corner of the current cell, and p_w and p_h represent the width and height of the predefined anchor box.

B. MOBILENETV3 BACKBONE REPLACEMENT

To address the bottlenecks of CSPDarknet53 in terms of computational complexity and power consumption, this paper replaces the original backbone network of YOLOv7 with MobileNetV3. The core motivation for this design is to significantly reduce the number of parameters and inference overhead without significantly sacrificing semantic expressive power. First, the CSPDarknet53 (Cross Stage Partial Darknet 53) backbone network in YOLOv7 is stripped out and replaced with the MobileNetV3 structure. MobileNetV3 consists of several MobileNetV3 modules, each of which contains a lightweight depthwise separable convolution, batch normalization, and the activation function H-Swish [17]. The core computational process of MobileNetV3 is represented as shown in formula (5).

$$Y = \sigma_{\text{H-Swish}}\left(\text{BN}(D_w(X))\right) \tag{5}$$

In formula (5), X represents the input feature map; D_w represents the depthwise convolution operation; and $\sigma_{\text{H-Swish}}$ represents the activation function. $\sigma_{\text{H-Swish}}$ is defined as shown in formula (6).

$$\sigma_{\text{H-Swish}}(x) = x \cdot \frac{\text{ReLU6}(x+3)}{6} \tag{6}$$

The MobileNetV3 module integrates a lightweight SE (Squeeze-and-Excitation) attention mechanism to enhance key information through channel attention. In SE, the channel descriptor formula calculated by global average pooling is shown in formula (7).

$$z_c = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} X_{i,j,c} \tag{7}$$

In formula (7), $X_{i,j,c}$ represents the pixel value at the (i, j)-th position in the c-th channel of the feature map.

Then, two fully connected layers are used to implement channel weight mapping and the final channel recalibration formula is shown in (8).

$$s = \sigma\left(W_2 \cdot \delta(W_1 \cdot z)\right), \quad \hat{X}_{i,j,c} = s_c \cdot X_{i,j,c} \tag{8}$$

In formula (8), W_1 and W_2 represent the dimensionality reduction and increase matrices, and σrepresents the ReLU activation. $\hat{X}_{i,j,c}$ represents the final channel recalibration expression.

C. CBAM ATTENTION INTEGRATION

In tasks such as identifying passenger line-of-sight areas and defects, targets are typically small in scale, have complex backgrounds, and are easily occluded. Relying solely on convolutional features can easily lead to them being overwhelmed by irrelevant backgrounds. To address this, this paper integrates a visual attention mechanism (CBAM) into the network neckline to perform explicit attention modeling of features. The visual attention mechanism (CBAM) selectively enhances the response of key channels and key spatial locations in the feature map by cascading channel attention and spatial attention mechanisms [18,19].

The role of CBAM in this paper is to dynamically recalibrate features based on channel and spatial dimensions. Its effectiveness depends on the input features containing sufficient semantic information (for distinguishing target categories/saliency) while retaining necessary multi-scale details (for locating small targets and handling occlusion). The network neck (using BiFPN) is responsible for "cross-scale fusion," integrating features from different depths (i.e., different receptive fields and scales) into a set of multi-scale representations according to learnable weights. Therefore, applying CBAM here can simultaneously adjust the channel importance and spatial response of each scale based on the fused context, thereby maximizing the signal-to-noise ratio and improving the recall and localization accuracy of subsequent decoding under conditions of small targets and high occlusion. In contrast, if CBAM is only placed in the last layer of the backbone, it can only access high semantic features at a single scale, lacking direct perception of low-level details and intermediatescale information, resulting in limited enhancement capabilities for small/dense targets. While extensive insertion of attention in multiple places in the backbone can compensate for this, it significantly increases computational and parameter overhead and destroys the stability of pre-trained features. In summary, considering scale-based semantic matching, computational overhead, and the stability of transfer learning, integrating CBAM into the neck is a more reasonable design choice.

In the visual attention mechanism (CBAM), channel attention generates two different channel descriptors through global average pooling and maximum pooling, as shown in formula (9).

$$\begin{aligned} \hat{f}^c_{\text{avg}} &= \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} \hat{F}(i,j,c), \\ \hat{f}^c_{\text{max}} &= \max_{i=1,\ldots,H;\, j=1,\ldots,W} \hat{F}(i,j,c). \end{aligned} \tag{9}$$

In formula (9), $\hat{F}(i,j,c)$ represents the pixel value of the feature map. In this paper, $\hat{f}^c_{\text{avg}}$ and $\hat{f}^c_{\text{max}}$ are input into a shared multilayer perceptron (MLP), as shown in formula (10).

$$M_{\text{avg}} = \text{MLP}(\hat{f}_{\text{avg}}), \quad M_{\text{max}} = \text{MLP}(\hat{f}_{\text{max}}) \tag{10}$$

After the channel attention weights are summed up by element-wise weighting, they are mapped to the [0, 1] interval using Sigmoid activation, and the channel-weighted feature map is output, as shown in (11).

$$M_c = \sigma(M_{\text{avg}} + M_{\text{max}}), \quad \hat{F}'(i,j,c) = M_c(c) \cdot \hat{F}(i,j,c) \tag{11}$$

In formula (11), $\hat{F}'$ represents the channel-weighted feature map, and M_c represents the channel attention weight.

In the spatial attention mechanism, the channel-weighted feature map is first subjected to cross-channel max pooling and average pooling to obtain two two-dimensional feature maps, as shown in formula (12).

$$\begin{aligned} \hat{f}^s_{\text{avg}}(i,j) &= \frac{1}{C}\sum_{c=1}^{C}\hat{F}'(i,j,c), \\ \hat{f}^s_{\max}(i,j) &= \max_{c=1,\ldots,C}\hat{F}'(i,j,c). \end{aligned} \tag{12}$$

Now the two are concatenated according to the channel dimension to form a feature map f^s, and a convolution layer is used to perform convolution operation to obtain the spatial attention weight and output the final spatial weighted feature map, as shown in (13).

$$M_s = \sigma\left(\text{Conv}_{k\times k}(\hat{f}^s)\right), \quad \hat{F}''_{i,j,c} = M_s(i,j,c)\cdot\hat{F}'_{i,j,c} \tag{13}$$

In formula (13), $\hat{F}''$ represents the final spatially weighted feature map, and M_s represents the spatial attention weight.

The Neck structure in YOLOv7 uses PANet for multi-scale feature fusion, and the CBAM module is embedded in the key nodes of PANet. Specifically, after feature fusion at each scale, CBAM processing is performed before being fed into the subsequent upsampling or downsampling layers.

D. SOFT-NMS DECODING ENHANCEMENT

In dense scenes (such as multiple people simultaneously viewing the same area or defects clustering on a conveyor belt), traditional NMS relies on a fixed IoU threshold for hard suppression of candidate boxes, which can easily lead to the erroneous deletion of real targets. To alleviate this problem, this paper introduces Soft-NMS in the decoding stage. Soft-NMS replaces hard threshold suppression by continuously decaying the confidence of the detection boxes [20].

Traditional NMS relies on a threshold and directly discards boxes whose intersection over union (IoU) exceeds the threshold. Soft-NMS, on the other hand, continuously attenuates the confidence of each box using a Gaussian function based on its overlap with the current highest confidence box.

The steps are as follows: (1) Sorting all boxes s_j in descending order of confidence. (2) Taking the box b_m with the highest confidence and adding it to the final detection result set. (3) Calculating the IoU between the remaining boxes b_j and b_m and update the confidence. (4) Deleting all boxes whose confidence s_j is lower than the threshold and repeat step (2) until the candidate box set is empty.

The Gaussian decay function formula is shown in (14).

$$s_j = s_j \times e^{-\frac{\text{IoU}(b_m,b_j)^2}{\beta}} \tag{14}$$

In formula (14), β represents the attenuation control parameter, with a value of 0.5.

E. BIFPN MULTI-SCALE FUSION

While PANet can achieve both top-down and bottom-up feature propagation, its indiscriminate treatment of features across all scales makes it difficult to adapt to the scale imbalance problem introduced by MobileNetV3. Therefore, this paper replaces PANet with a BiFPN structure to achieve learnable multi-scale weighted fusion. BiFPN achieves efficient information flow and feature fusion through weighted bidirectional connections, enhancing the expressive power of features at different scales.

The multi-scale feature maps extracted from Backbone correspond to different resolution levels and serve as input features. BiFPN introduces learnable weights to perform weighted summation of input features from different paths to avoid the information loss caused by simple mean fusion. The weights are activated by ReLU to ensure non-negativity. The calculation formula is shown in (15).

$$\hat{w}_i = \max(0, w_i), \quad w_i = \frac{\hat{w}_i}{\sum_j \hat{w}_j + \epsilon} \tag{15}$$

In formula (15), w_i represents the learnable weight, and ϵ is 10^{-4}.

The weighted fusion result is shown in formula (16).

$$\tilde{F} = \sum_i w_i \cdot X_i \tag{16}$$

In formula (16), X_i represents input features of different scales.

BiFPN realizes bottom-up and top-down feature flow, and after the features of each node are fused, dimensionality reduction and non-linear mapping are performed through Depthwise Separable Convolution (DWConv), as shown in formula (17).

$$\tilde{F}_{\text{out}} = \text{DWConv}(\tilde{F}) + \tilde{F} \tag{17}$$

In formula (17), $\tilde{F}_{\text{out}}$ represents the output after dimensionality reduction and non-linear mapping. The lightweight optimized YOLOv7 and CBAM structures are shown in Figure 1.

In Figure 1, based on the input image, the lightweight and optimized YOLOv7 architecture constructed in this paper uses MobileNetV3 as the backbone network, replacing the original YOLOv7′s CSPDarknet. After the backbone outputs features, the CBAM attention module is introduced to apply channel attention and spatial attention in sequence to enhance the model's perception of key targets such as the passenger's gaze area. The Neck structure uses BiFPN to fuse feature maps of different scales and combines them with Soft-NMS decoding to suppress redundant boxes.

F. PASSENGER GAZE TARGET MAPPING

To investigate how to calculate the gaze from camera-generated passenger facial and eye images through eye tracking and spatially map it to the bounding boxes of interface elements detected by YOLOv7 to determine the

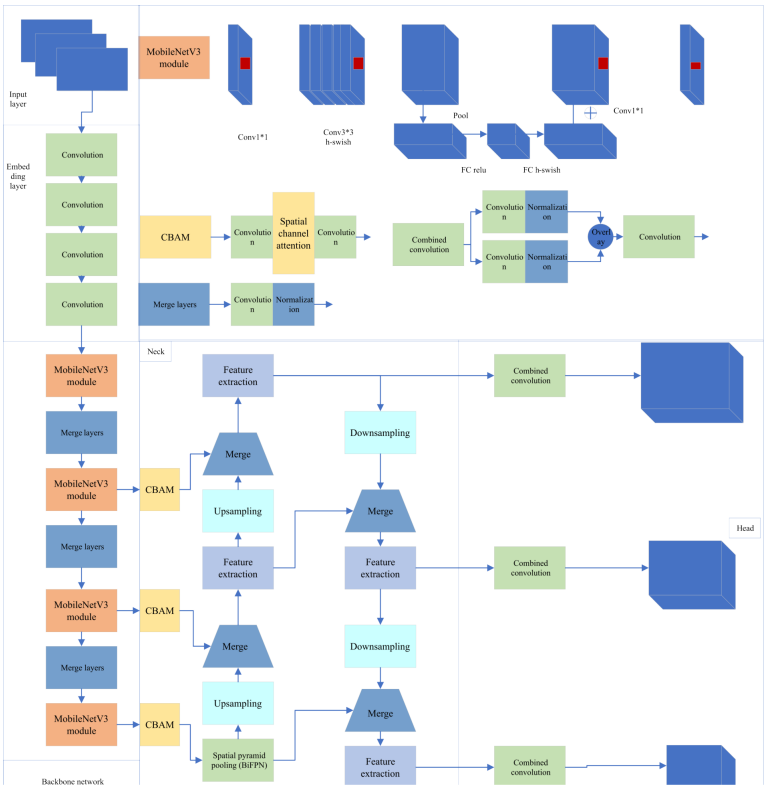

FIGURE 1. Lightweight optimized YOLOv7 and CBAM structure

target the passenger is looking at, this paper proceeds through the following steps: 1) eye region detection and 3D pupil model fitting; 2) gaze direction vector estimation; 3) intersection of the gaze and the screen plane; and 4) matching the intersection point with the detected object bounding box.

1) Eye Region Detection and 3D Pupil Model Fitting

In this paper, for an input facial image frame, the facial keypoint detector MediaPipe is used to extract the 2D pixel coordinates $\{(\zeta_i, \upsilon_i)\}_{i=1}^{M}$ of the eye contours and pupil center [21,22]. A standard 3D eye model point set is then used, where each point $\iota_i = (\iota_{xi}, \iota_{yi}, \iota_{zi})$ is the reference position of the pupil and corneal edge in the eyeball coordinate system. Finally, the camera pose $[R\ t]$ is solved using the least squares PnP method, satisfying the formula (18) [23,24]. Formula (18) is then iteratively optimized and minimized.

$$\xi \begin{bmatrix} \zeta_i \\ \upsilon_i \\ 1 \end{bmatrix} = \pi [R\ t] \begin{bmatrix} \iota_{xi} \\ \iota_{yi} \\ \iota_{zi} \\ 1 \end{bmatrix} \tag{18}$$

In formula (18), π represents the camera intrinsic parameter matrix, and ξ represents the scaling factor. This paper presents a clear 2D–3D registration and robust solution based on facial details extracted by MediaPipe Face Mesh. First, MediaPipe is used to obtain 468 facial keypoints per frame and iris keypoints for each eye (5 points are commonly used). From these, a set of two-dimensional pixels, including the corner of the eye, the eyelid edge, and the iris center, are selected as observations. Simultaneously, a standardized 3D eye model (including the pupil center, corneal apex, and several reference points) is used as the corresponding three-dimensional reference point set. Using the camera-integrated distortion coefficients obtained beforehand through a checkerboard or calibration board, the input image is distorted. Then, solvePnP (robust solution with RANSAC and subsequent refinement based on iterative least squares/Levenberg–Marquardt) is used to solve for the rotation matrix R and translation vector t to obtain the pose from the eye coordinate system to the camera coordinate system. The actual gaze vector is obtained by transforming the reference gaze direction (such as the reference vector pointing to the lens) in the eye coordinate system by R, and then intersecting it with a pre-estimated or manually calibrated screen plane (determined by the positions of at least 4 screen corner points in the camera coordinate system) to obtain the gaze intersection point on the screen plane; this intersection point is then projected back to the pixel plane and matched with the bounding boxes of interface elements detected by YOLOv7. To improve robustness, the system incorporates engineering processing in two places: (1) setting a threshold (default 0.5) based on the confidence of key points returned by MediaPipe, degenerating to a hybrid estimation of head pose + historical trajectory prediction when the confidence is low; (2) applying exponential weighted smoothing (0.6) and real-time error monitoring

to each frame gaze point (reprojection error exceeding ~2 pixels triggers re-estimation or discard), and supporting optional 5-point user calibration to eliminate individualized bias.

2) Estimation of the Gaze Direction Vector

The gaze reference vector $d_0 = [0, 0, 1]^T$ is defined in the eyeball coordinate system, pointing in the direction of the camera. The actual gaze direction is calculated by applying the rotation matrix, as shown in (19).

$$\hat{v} = Rd_0 \tag{19}$$

In formula (19), $\hat{v}$ represents the actual line of sight direction, and R represents the rotation matrix obtained by solving the PnP formula.

3) Intersection of Line of Sight and Screen Plane

Assuming that the plane of the urban rail guide screen satisfies the plane formula in the camera coordinate system, and the line of sight acquisition formula is as shown in formula (20).

$$\begin{aligned} \tau^T \phi + d_{\text{is}} = 0, \\ \phi(\varphi) = \chi - \varphi\hat{v}, \quad \varphi > 0 \end{aligned} \tag{20}$$

In formula (20), τ represents the plane normal; d_{is} represents the plane intercept; and ϕ represents the coordinates of the spatial point. χ represents the center position of the eyeball, which is equal to t. Solving for the intersection parameters so that $\phi(\varphi^*)$ falls on the screen plane and calculate the intersection coordinates, as shown in formula (21).

$$\begin{aligned} \tau^T(\chi - \varphi^*\hat{v}) + d_{\text{is}} = 0 \Rightarrow \varphi^* = \frac{\tau^T\chi + d_{\text{is}}}{\tau^T\hat{v}}, \\ \phi^* = \chi - \varphi^*\hat{v} \end{aligned} \tag{21}$$

In formula (21), φ^* represents the intersection parameter, and ϕ^* represents the intersection coordinate.

4) Intersection Coordinate Conversion and Target Matching

Now projecting the intersection point ϕ^* back onto the image plane to obtain the screen pixel coordinates (ζ^*, v^*). The set of interface target frames detected by YOLOv7, where each frame is $[x_j^{\min}, y_j^{\min}, x_j^{\max}, y_j^{\max}]$. In the attention target matching, each detected interface target frame is traversed.

If the following formula (22) is satisfied, the interface element is determined to be the passenger's attention target; if multiple frames are hit, the frame with the largest coverage area is selected first.

$$x_j^{\min} \le \zeta^* \le x_j^{\max} \quad \wedge \quad y_j^{\min} \le v^* \le y_j^{\max} \tag{22}$$

In this paper, the output of YOLOv7/ByteTrack is not used to infer the gaze direction itself, but only as a temporally consistent spatial constraint and identity maintenance mechanism to provide stable input for PnP-based geometric gaze estimation. In each frame, the detection and tracking module is only responsible for continuously and reliably locating and identifying the head/face region (track ID) of the same passenger in the temporal dimension, thereby assigning a consistent individual identity across frames to the eye keypoints extracted by MediaPipe; subsequently, the actual gaze direction vector is obtained entirely from the eye key points corresponding to this identity and the standard 3D eye model through PnP geometric solution, rather than inferred from bounding box regression or motion information. Furthermore, in the temporal series, the track IDs provided by ByteTrack are used as a "container" for the PnP solution results, that is, the gaze vector and the screen intersection under the same track ID are continuously associated, and temporal smoothing and short-term occlusion compensation are performed through NSA Kalman filtering, so as to maintain the continuity and consistency of the gaze target ID even when the keypoints in individual frames are unstable or partially missing. The method presented in this paper forms a collaborative framework in which "detection/ tracking is used for identity and temporal consistency, PnP is used for geometric gaze estimation, and the two are decoupled in the temporal domain but coupled at the ID level," which ensures the traceability and physical interpretability of the gaze target in consecutive frames.

IV. IMPROVED BYTETRACK TRACKING

A. ORIGINAL BYTETRACK ALGORITHM

To achieve efficient and robust multi-target tracking, this paper constructs a tracking framework for passenger gaze targets based on the ByteTrack algorithm. ByteTrack effectively handles occlusion and target loss scenarios by decoupling high-confidence and low-confidence detection frames [25,26].

1) Target Detection Frame Confidence Classification

Let $D_t = \{d_i\}_{j=1}^{N_t}$ be the set of all detection boxes in frame t of the video. Each detection box d_i consists of bounding box coordinates and a confidence score s_i, and N_t is the number of detection boxes. ByteTrack first divides the detection boxes into a high-confidence set $D_t^h = \{d_i \mid s_i \ge \theta\}$ and a low-confidence set $D_t^l = \{d_i \mid s_i < \theta\}$ based on the confidence threshold, where θ is 0.6.

2) Kalman Filter Prediction State

For the set of all tracked trajectories in frame $t-1$, the state vector of each trajectory is defined as shown in formula (23).

$$x_j = [x_j, y_j, w_j, h_j, \dot{x}_j, \dot{y}_j, \dot{w}_j, \dot{h}_j]^T \tag{23}$$

In formula (23), x_j, y_j, w_j, and h_j represent the center coordinates and dimensions of the bounding box, and $\dot{x}_j$, $\dot{y}_j$, $\dot{w}_j$, and $\dot{h}_j$ represent the corresponding velocity components.

The Kalman filter uses a linear dynamic model to predict the trajectory state, as shown in formula (24).

$$\begin{aligned} x_j^{t|t-1} &= Fx_j^{t-1|t-1} + u, \\ P_j^{t|t-1} &= FP_j^{t-1|t-1}F^T + Q \end{aligned} \tag{24}$$

In formula (24), F represents the state transfer matrix; u represents the control vector; P represents the state covariance matrix; and Q represents the process noise covariance.

3) Hungarian Algorithm Data Association

ByteTrack associates the set of high-confidence detection boxes with the predicted trajectory state using the Hungarian algorithm. The elements in the cost matrix are defined as negative IOU distances, as shown in (25).

$$\hat{C}_{j,i} = 1 - \mathrm{IoU}(b_j^{t|t-1}, b_i) \tag{25}$$

In formula (25), $\hat{C}_{j,i}$ represents the cost matrix, and $b_j^{t|t-1}$ represents the trajectory prediction box. By

solving the Hungarian algorithm to minimize the total cost, a set of matching pairs is obtained.

4) Trajectory Status Update and Low-Confidence Detection Box Verification

For successfully matched trajectories, a Kalman filter measurement update is applied. For unmatched trajectories, a set of low-confidence detection boxes is used to perform matching again. The cost matrix is constructed as above, and the Hungarian algorithm is executed to obtain a matching set. The status of the matched trajectories is then updated.

In track management, tracks that have been lost for more than a preset number of frames are deleted, and new tracks are initialized for newly detected, unmatched high-confidence detection boxes. A track lifecycle mechanism is used to prevent mismatches and track drift. ByteTrack uses thresholding to prioritize tracking of high-confidence targets, with low-confidence targets serving as auxiliary tracking, mitigating missed detections and mismatches caused by occlusion. It also uses an IOU threshold to optimize the matching strategy, improving tracking continuity in occluded scenarios.

B. NSA KALMAN FILTER PREDICTION

To improve the prediction accuracy of the passenger's gaze target motion state, this paper introduces the NSA Kalman filter to replace the traditional linear Kalman filter [27,28]. The NSA Kalman filter dynamically adjusts the filter gain by online estimating the system noise covariance to adapt to the non-linear motion characteristics caused by changes in passenger behavior.

The target state vector is shown in formula (26).

$$x_k = [x_k, y_k, w_k, h_k, \dot{x}_k, \dot{y}_k, \dot{w}_k, \dot{h}_k]^T \tag{26}$$

In formula (26), k represents the time step.

The state transition formula and observation model are shown in formula (27).

$$\begin{aligned} x_k &= Fx_{k-1} + u_{k-1} + w_{k-1}, \\ z_k &= Hx_k + v_k \end{aligned} \tag{27}$$

In formula (27), H represents the observation matrix.

In traditional Kalman filtering, the process noise covariance Q and the observation noise covariance R are usually fixed values, making it difficult to adapt to the non-linear and dynamic changes in passenger behavior. The NSA Kalman filter achieves adaptive adjustment by estimating Q_k and R_k in real time. The update formula is shown in formula (28).

$$\begin{aligned} Q_k &= \gamma Q_{k-1} + (1-\gamma)(K_k S_k K_k^T), \\ R_k &= \delta R_{k-1} + (1-\delta)(v_k v_k^T) \end{aligned} \tag{28}$$

In formula (28), γ and δ represent smoothing coefficients; K_k represents the Kalman gain; S_k represents the innovation covariance; and v_k represents the measurement residual.

The Kalman gain is calculated as shown in formula (29).

$$K_k = P_{k|k-1}H^T(HP_{k|k-1}H^T + R_k)^{-1} \tag{29}$$

The prediction covariance is updated as shown in formula (30).

$$P_{k|k-1} = FP_{k-1|k-1}F^T + Q_k \tag{30}$$

C. OPTIMIZATION OF GIOU MATCHING STRATEGY

To improve the accuracy of multi-target association, especially tracking continuity under severe occlusion, this paper replaces the traditional IoU-based matching strategy with the GIoU matching method.

Given the predicted bounding box and the detected bounding box, the IoU is defined as shown in formula (31).

$$\mathrm{IoU}(B_p, B_d) = \frac{|B_p \cap B_d|}{|B_p \cup B_d|} \tag{31}$$

In formula (31), B_p represents the predicted bounding box and B_d represents the detected bounding box. When the two boxes have no intersection, IoU is zero, which cannot reflect the relative position relationship of the boxes, resulting in the failure of the matching judgment.

GIoU introduces the minimum containing box η, which is defined as shown in formula (32).

$$\mathrm{GIoU}(B_p, B_d) = \mathrm{IoU}(B_p, B_d) - \frac{|\eta \setminus (B_p \cup B_d)|}{|\eta|} \tag{32}$$

In formula (32), $|\eta \setminus (B_p \cup B_d)|$ represents the area difference between the containing box and the union of the two boxes, and $|\eta|$ represents the area of the containing box.

The matching cost matrix is constructed as shown in formula (33).

$$O_{i,j} = 1 - \mathrm{GIoU}(B_{p_i}, B_{d_j}) \tag{33}$$

In formula (33), $O_{i,j}$ represents the matching cost matrix.

The real-time detection and tracking framework for the passenger's gaze target is shown in Figure 2.

In Figure 2, the system uses the image stream obtained by the camera as input. It detects candidate targets through a lightweight YOLOv7 that integrates the MobileNetV3 backbone network and the CBAM attention mechanism, and improves detection accuracy and stability through BiFPN feature fusion and Soft-NMS decoding. Then, the facial and eye key points are located, combined with PnP posture solution and line of sight intersection mapping technology to achieve accurate determination of the gaze target.

V. PASSENGER GAZE TARGET DETECTION AND REAL-TIME TRACKING EXPERIMENT

A. EXPERIMENTAL DATA AND PREPROCESSING

This study collected passenger gaze data for six months (Jan–Jun 2025) at four Beijing subway stations (e.g., Capital Airport Terminal 2) using a top-mounted industrial camera Model MV CA050 20GC with resolution 1920 by 1080 pixels and frame rate 30 frames per second, capturing 180000 frames across various passenger densities, lighting, and layouts. The total raw data was uniformly sampled at 2 frame intervals to yield 15000 keyframes. Using the LabelImg tool, 20 categories of guide interface elements, valid facial rectangles, and pupil centers were annotated and cross-checked. Three trained annotators performed the annotation, and inter annotator agreement was evaluated using Cohen's Kappa statistic. The data was split using 5 fold cross validation with 80 percent of keyframes for training and 20 percent for validation in each fold, and an independent test set comprising 10 percent of the total keyframes was held out before cross validation.

B. EVALUATION METRICS

MOTA :

$$\mathrm{MOTA} = 1 - \frac{\sum_t (\mathrm{FN}_t + \mathrm{FP}_t + \mathrm{IDS}_t)}{\sum_t \mathrm{GT}_t} \tag{34}$$

In formula (34), FN_t (False Negatives) represents the number of missed targets, FP_t (False Positives) represents the number of misdetected targets, IDS_t (Identity Switches) represents the number of ID switches, and GT_t (Ground Truth) represents the number of true targets.

MOTP (Multiple Object Tracking Precision):

$$\mathrm{MOTP} = \frac{\sum_{t,i} \omega_t^{(i)}}{\sum_t \psi_t} \tag{35}$$

In formula (35), $\omega_t^{(i)}$ represents the distance to the matching target, and ψ_t represents the number of successfully matched targets.

IDF1 (Identity F1 Score):

$$\mathrm{IDF1} = \frac{2\,\mathrm{IDTP}}{2\,\mathrm{IDTP} + \mathrm{IDFP} + \mathrm{IDFN}} \tag{36}$$

In formula (36), IDTP (Identity True Positives) represents the number of frames with correct identity matching, IDFP (Identity False Positives) represents the number of frames with misidentified identity, and IDFN (Identity False Negatives) represents the number of frames with no identity matching.

TABLE 1. Experimental parameters

Parameters	Value	Parameters	Value
Input image size	640 × 640	Bounding box loss weight	3.0
Number of training rounds	300	Confidence loss weight	1.0
Batch size	32	GIoU loss weight	1.5
Initial learning rate	0.001	Motion state prediction error loss	2.0
Class loss weight	1.0	Appearance feature matching loss	1.0

C. EXPERIMENTAL DESIGN

The core of this study's passenger gaze system is the optimized YOLOv7-CBAM model. In the object detection experiment, its adaptability and accuracy (mAP@0.5) were verified against mainstream models like Faster R-CNN, YOLOv5m, YOLOv8m, and lightweight variants (YOLOv8-MobileNetV3, YOLOv7-ShuffleNetV1, YOLOv7- GhostNet) under a unified environment and data strategy, specifically targeting scenarios with limited computing power. In the subsequent tracking experiment, the detection outputs were integrated with multiple mainstream trackers—including SORT, DeepSORT, OC-SORT, BoT-SORT, and StrongSORT—for comparison with the improved ByteTrack algorithm. The performance was comprehensively evaluated using MOTA, MOTP, and IDF1 to confirm the robustness and real-time capability of the improved strategy against multi-target occlusion, rapid motion, and weak detection signals, with final verification focusing on FPS and performance under varying occlusion levels.

The experimental parameter settings are shown in Table 1.

VI. PASSENGER ATTENTION TARGET DETECTION AND REAL-TIME TRACKING EVALUATION

The gaze target is located in the current frame of the subway station model frame and tracked in real-time to the next frame.

A. CBAM ATTENTION FEATURE MAP DISPLAY

The CBAM attention feature map is shown in Figure 3.

Figure 3 shows the comparison of the feature maps of the lightweight YOLOv7 model before and after the application of the CBAM attention mechanism. Figure 3 (a) is the feature map extracted by the original YOLOv7. It can be seen that its response area distribution is relatively uniform and lacks discrimination. Figure 3(b) shows that after introducing the channel attention module of CBAM, the feature response is strengthened in the direction of the significant channel, which enhances the semantic expression ability of the key area. Figure 3(c) shows that after further integrating the spatial attention mechanism, the model forms

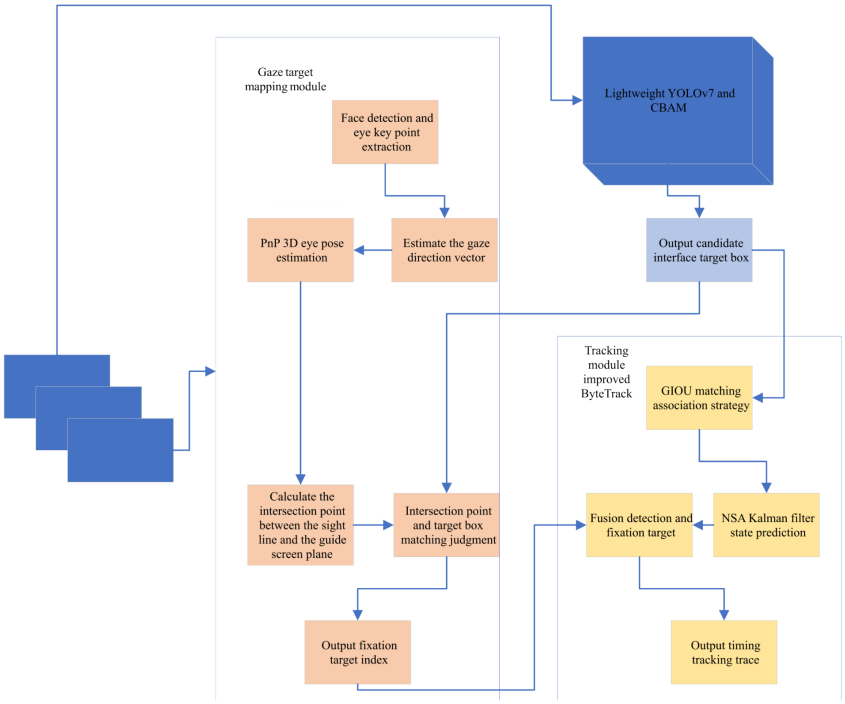

FIGURE 2. Real-time detection and tracking framework of passenger gaze targets

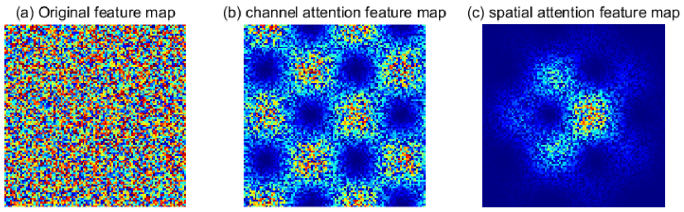

FIGURE 3. CBAM attention feature map display

an obvious high-response area in the passenger's gaze area (guide interface, screen interaction point, etc.), and the nonattention area is effectively suppressed.

B. DETECTION ACCURACY EVALUATION

The detection accuracy evaluation results are shown in Figure 4.

The detection accuracy evaluation results show that the performance of different models at different stations varies significantly. In Figure 4(a), at the Capital International Airport Terminal 2 station, the YOLOv7-CBAM model achieves the best mAP@0.5 of 96.3% and F1-score of 96.5%, which are approximately 3.2% and 3.4% higher than the YOLOv8m model's 93.1% mAP and 93.1% F1-score. The lightweight version of YOLOv8 (MobileNetV3) achieves only 86.5% mAP and 86.6% F1-score, which is worse than YOLOv8m, reflecting the significant impact of different backbones on small object detection. In the Guomao Station scenario in Figure 4(b), the overall accuracy is slightly lower than that of Capital Airport. The YOLOv7-CBAM model achieves a mAP@0.5 of 95.4% and an F1-score of 95.4%, a 5.4% improvement over YOLOv8m's 90.0% mAP and 90.0% F1-score. Faster R-CNN achieves only 79.9% mAP@0.5 and 79.6% F1-score for this site.

In Figure 4(c) at Tiananmen East Station, the YOLOv7-CBAM model achieves a mAP@0.5 score of 96.0% and an F1-score of 95.8%. In contrast, other models such as YOLOv5m and YOLOv8m achieve scores of 85.5%/85.5% and 88.0%/88.1%, respectively, which are significantly lower than the optimized model. In particular, Faster R-CNN's mAP@0.5 drops to 75.8% and its F1-score to 75.7% at this station, indicating its increased sensitivity to lighting and environmental changes. In Figure 4(d) of Wudaokou Station, under the most crowded conditions, the optimized YOLOv7-CBAM model still maintains a 95.6% mAP@0.5 and 95.5% F1-score, while YOLOv7 (GhostNet) achieves 90.5% and 90.5%. This method represents a 10% improvement over the lightweight YOLOv8 (MobileNetV3) model's 85.3% and 85.2%.

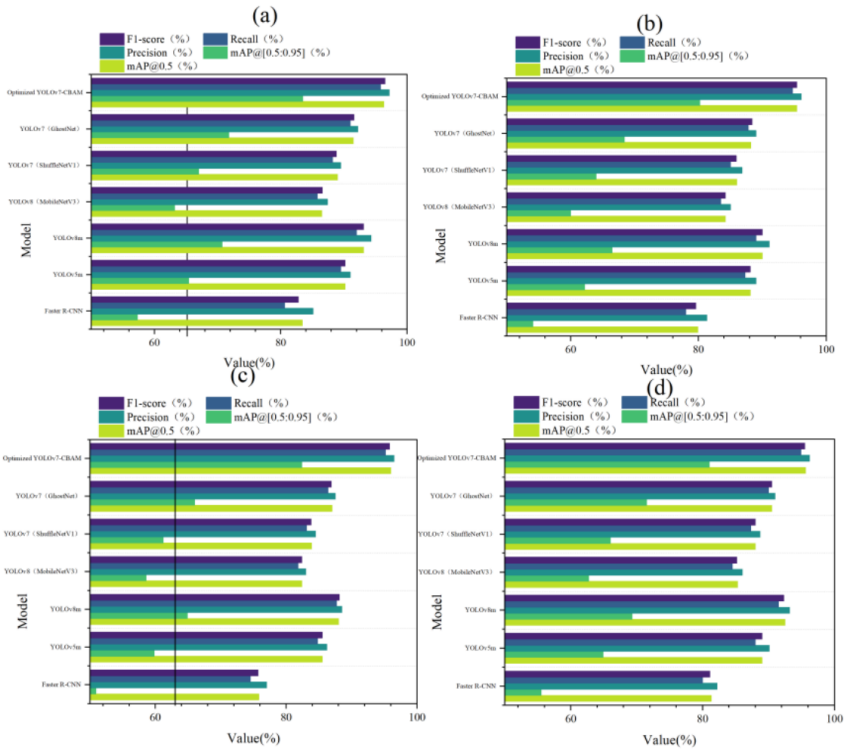

FIGURE 4. Detection Accuracy Evaluation: (a) Capital Airport Terminal 2 Station; (b) Guomao Station; (c) Tiananmen East Station; (d) Wudaokou Station

C. TRACKING ACCURACY EVALUATION

The tracking accuracy evaluation is shown in Figure 5.

In Figure 5(a), at Terminal 2 of Beijing Capital International Airport, both occlusion and crowd density are high. SORT's IDF1 is 68.3% and MOTA is 58.2%, the weakest performance. DeepSORT improves to 75.5%/65.8%, OC-SORT further increases to 81.2%/71.0%, BoT-SORT reaches 85.1%/76.5%, and StrongSORT reaches 89.4%/81.9%. The improved ByteTrack algorithm achieves 94.2% IDF1 and 88.3% MOTA in this scenario, maintaining a MOTP (Motability Percentage) of 85.7%. Compared to StrongSORT, these improvements represent an increase of 4.8%, 6.4%, and 0.1% in IDF1, MOTA, and MOTP (Motability Percentage), respectively. This demonstrates extremely high identity consistency and positioning accuracy. Figure 5(b) shows a similarly dense crowd at Guomao Station, but with some occlusion. SORT achieves an IDF1 of 63.5% and a MOTA of 53.1%. DeepSORT increases this to 71.3%/61.4%, OC-SORT to 76.5%/66.2%, BoT-SORT to 80.8%/72.1%, and StrongSORT to 84.2%/75.5%. Figure 5(c) shows that the complex lighting and viewing angles at Tiananmen East Station lead to a general decline in overall tracking performance. SORT achieves an IDF1 of 58.4% and a MOTA of 48.8%. DeepSORT increases to 67.0%/57.2%, OC-SORT to 73.1%/64.0%, BoT-SORT to 78.0%/70.8%, and StrongSORT to 83.7%/75.3%. The improved ByteTrack algorithm achieves 89.8% IDF1, 82.1% MOTA, and 82.7% MOTP, maintaining robust tracking capabilities under strong lighting variations. Figure 5(d) shows relatively concentrated streamlines and evenly illuminated areas at Wudaokou Station, resulting in excellent overall tracking performance. SORT achieves 71.7%/61.9% in terms of IDF1 and MOTA; DeepSORT achieves 77.2%/68.8%; OC-SORT achieves 82.5%/74.2%; BoT-SORT achieves 86.3%/78.0%; and StrongSORT achieves 90.1%/83.4%. The improved ByteTrack algorithm achieves 93.7% IDF1, 87.0% MOTA, and 85.2% MOTP, performing even better in less extreme occlusion scenarios.

D. TRACKING REAL-TIME EVALUATION

This paper describes the end-to-end deployment and measurement of the model on a representative edge platform: NVIDIA Jetson AGX Xavier development kit (32GB system memory, configured for 20W power consumption). Performance measurements were performed using end-to-end timing (including image preprocessing, detection, gaze estimation, data association, and post-processing), with each experiment continuously inferenced for at least 3000 frames on the same test set and repeated 5 times for averaging. All timing and power results are reported as mean $\pm$ standard deviation from five independent runs (n=5). Power consumption was sampled and averaged during steady-state conditions using an external power meter (with onboard current sensor recording for cross-validation).

To evaluate the real-time performance of various tracking solutions on edge devices, the experiment took Terminal 2 of Beijing Capital International Airport as an example, focusing on FPS, model parameter count, end-to-end response,

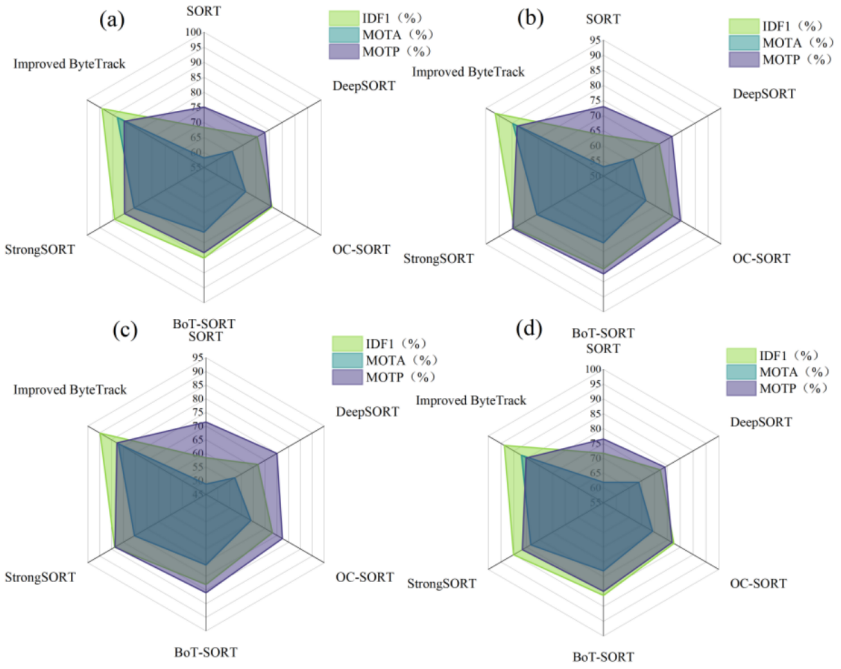

FIGURE 5. Tracking Accuracy Evaluation: (a) Capital Airport Terminal 2 Station; (b) Guomao Station; (c) Tiananmen East Station; (d) Wudaokou Station

and pure inference time, as well as GPU (Graphics Processing Unit) and CPU (Central Processing Unit) resource usage. The results are shown in Figure 6.

In Figure 6(a), the response and inference times of the optimized YOLOv7-CBAM + Improved ByteTrack at Terminal 2 of Beijing Capital International Airport show an end-to-end response time of only 30.4ms and a pure inference time of 26.3ms, both outperforming other solutions. Faster R-CNN + ByteTrack achieves response and inference times of 81.5/72.1ms, while YOLOv8 (MobileNetV3) + DeepSORT achieves 43.2/35.6ms, and YOLOv7 (ShuffleNetV1) + Improved ByteTrack achieves 33.6/28.4ms. This shows that the introduction of CBAM and a lightweight backbone network effectively reduces overall latency. In terms of GPU and CPU resource utilization in Figure 6(b), Faster R-CNN + ByteTrack occupies 85.2% of the GPU and 18.7% of the CPU, putting the greatest strain on resources. YOLOv8 (MobileNetV3) + DeepSORT occupies 47.9% of the GPU and 34.5% of the CPU, YOLOv7 (ShuffleNetV1) + Improved ByteTrack occupies 36.3% and 24.8% respectively, and YOLOv7 (GhostNet) + Improved ByteTrack occupies 52.1% and 29.6% respectively. The optimized YOLOv7-CBAM model combination achieves a good GPU utilization of 41.7% and CPU utilization of 22.4%, achieving a good balance between hardware utilization and performance.

Judging from the FPS and model parameter count in Figure 6(c), the optimized solution achieves 32.2 FPS with only 19.7M parameters; Faster R-CNN + ByteTrack achieves 12.3 FPS/58.4M, YOLOv8 (MobileNetV3) + DeepSORT achieves 24.7 FPS/26.8M, YOLOv7 (ShuffleNetV1) + Improved ByteTrack achieves 29.8 FPS/14.5M, and YOLOv7 (GhostNet) + Improved ByteTrack achieves 26.7 FPS/24.9M, proving that the introduction of the lightweight backbone and attention module achieves higher real-time processing capabilities while maintaining a smaller model size.

TABLE 2. T-test results

Comparison Model	t value	p value	Significance
Faster R-CNN + ByteTrack	18.4	< 0.0001	***
YOLOv8 (MobileNetV3) + DeepSORT	6.7	0.00002	***
YOLOv7 (ShuffleNetV1) + ByteTrack	3.9	0.0031	**
YOLOv7 (GhostNet) + ByteTrack	4.8	0.0007	***

Note: ***$p < 0.001$, **$p < 0.01$.

The experiment performs a paired sample t-test on the FPS of each model as follows:

(1) Null hypothesis (H_0): The average FPS of the optimized YOLOv7-CBAM + improved ByteTrack is not significantly different from the average FPS of the comparison model.

(2) Alternative hypothesis (H_1): There is a significant difference in the average FPS between the two.

(3) For each pair of models (optimized versus a baseline), five independent runs are performed, and FPS results are reported as mean ± standard deviation (n=5); the difference sequence is calculated for paired t-test.

(4) The mean and standard deviation of the difference are calculated, and the t value and p value are

calculated.

(5) If p<0.05), H is rejected and it is considered that there is a significant difference in the FPS between the two. The t-test results are shown in Table 2.

In Table 2, the paired sample t-test results show that the optimized YOLOv7-CBAM + improved ByteTrack has

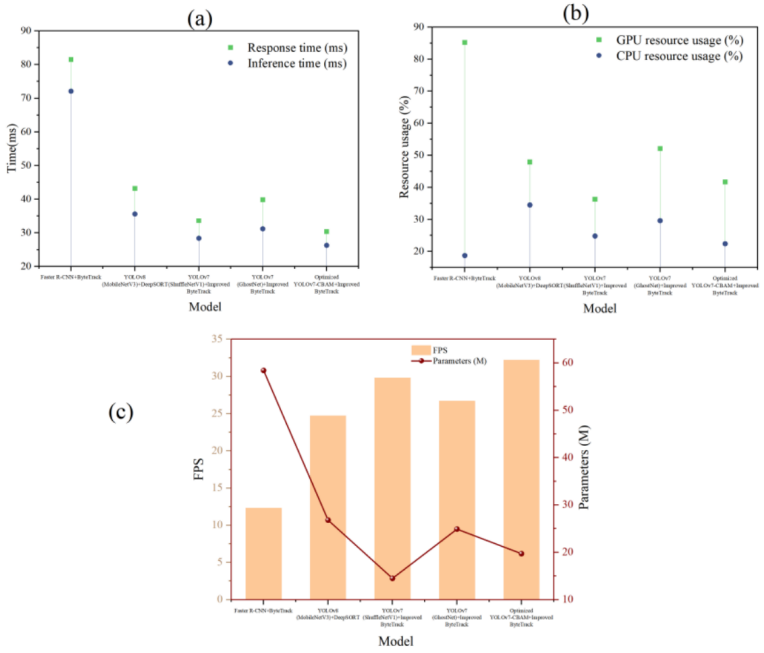

FIGURE 6. Real-time tracking evaluation: (a) Response time and inference time; (b) GPU and CPU resource utilization; (c) FPS and parameter count

highly significant differences in FPS performance compared with all baseline models. The difference with FasterR-CNN + ByteTrack is most significant (t=18.4, p<0.0001), and the difference with YOLOv8 (MobileNetV3) + DeepSORT is still extremely significant (t=6.7, p=0.00002). The optimized YOLOv7-CBAM + improved ByteTrack rejects the null hypothesis compared to both YOLOv7 (GhostNet) + ByteTrack (t=4.8, p=0.0007) and ShuffleNetV1 (t=3.9, p=0.0031), both using lightweight backbones. This fully demonstrates the statistical reliability of our method in improving processing speed.

Power consumption analysis is shown in Table 3.

Table 3 shows that Faster R-CNN + ByteTrack has the highest average power consumption of 28.7W, with a single-frame energy consumption of 2.33J. Meanwhile, the lightweight YOLOv7 (ShuffleNetV1) + ByteTrack has the lowest average power consumption of only 18.9W, with a single-frame energy consumption of 0.63J. YOLOv8 (MobileNetV3) + DeepSORT and YOLOv7 (GhostNet) + ByteTrack consume 22.4W/0.91J and 24.1W/0.90J, respectively. The optimized YOLOv7-CBAM + improved ByteTrack achieves moderate power consumption (20.7W) and low single-frame energy consumption (0.64J) while maintaining real-time performance of approximately 32.2 FPS. This demonstrates that the strategy of integrating a lightweight backbone with an attention mechanism strikes a good balance between power consumption and performance.

E. GAZE MAPPING ACCURACY

The experiment takes Terminal 2 of Beijing Capital International Airport as an example to explore the mapping accuracy under different gaze angles, lighting changes, and partial occlusion conditions. The results are shown in Figure 7.

Figure 7(a) shows that as the gaze angle gradually shifts from frontal (0°) to 90° sideways, the accuracy decreases from 97.2% to 81.3%, demonstrating that gaze deviation significantly impacts the model's localization capabilities. Figure 7(b) further illustrates the performance variations under different lighting conditions. The model performs best under normal lighting (96.5%), while the accuracy drops to 88.4% and 90.7% under strong glare and backlight, respectively, indicating that extreme lighting conditions can affect feature extraction in key areas.

Figure 7(c) explores the model's robustness under partial occlusion. The highest accuracy (97.0%) is achieved with no occlusion. However, the accuracy is lower when there is hair occlusion, glare from glasses, a hat brim on the upper edge, or partial mask occlusion, with the lowest accuracy reaching 87.6%, corresponding to the presence of a partial mask. Overall accuracy remains high under certain levels of occlusion, indicating that the introduction of the attention mechanism can mitigate environmental interference to some extent. However, different types of occlusion do pose challenges to the model's gaze point recognition accuracy, particularly occlusion of the eye and facial regions.

F. ABLATION EXPERIMENTS

An incremental ablation experiment was designed, using the complete scheme (MobileNetV3 + CBAM + BiFPN + Gaussian Soft-NMS + improved ByteTrack) as the baseline. Key components were removed/replaced (CBAM removed, BiFPN replaced with PANet, Soft-NMS replaced with hard NMS, MobileNetV3 replaced back to the original YOLOv7

TABLE 3. Power consumption analysis results

Model	Average power consumption (W)	Energy consumption per frame (J/frame)
Faster R-CNN + ByteTrack	28.7	2.33
YOLOv8 (MobileNetV3) + DeepSORT	22.4	0.91
YOLOv7 (ShuffleNetV1) + ByteTrack	18.9	0.63
YOLOv7 (GhostNet) + ByteTrack	24.1	0.90
Optimized YOLOv7-CBAM + improved ByteTrack	20.7	0.64

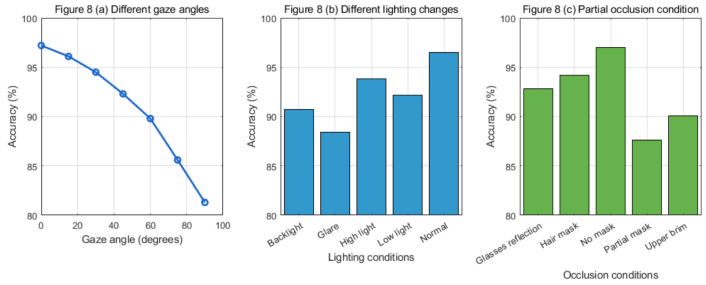

FIGURE 7. Gaze point mapping accuracy: (a)Different gaze angles; (b) Different lighting conditions (c)Partial occlusion

backbone CSPDarknet53, and the improved ByteTrack replaced with the original ByteTrack (linear Kalman + IoU)). Training and testing were repeated under the same training/validation partitions and hyperparameters. Key detection and tracking metrics (mAP@0.5, MOTA, IDF1), system real-time performance (FPS), and power consumption (average power consumption) were recorded to quantify the contribution of each component to the overall system performance and real-time performance. The ablation experiment results are shown in Table 4.

Table 4 clearly quantifies the contribution of each key component to system performance and efficiency through ablation experiments. Removing CBAM significantly reduced mAP@0.5 from 96.3% to 93.5%, while MOTA and IDF1 decreased by 3.8 and 3.5 percentage points, respectively, indicating that the attention mechanism plays a crucial role in detection accuracy and identity consistency in complex occlusion and small target scenes. Furthermore, replacing BiFPN with PANet reduced mAP@0.5 to 94.5% and MOTA to 86.7%, validating the advantage of weighted multi-scale feature fusion in consistently improving detection and tracking performance. In contrast, replacing Gaussian Soft-NMS with Hard NMS slightly increased FPS to 33.0, but reduced mAP and IDF1 to 95.7% and 93.0%, respectively, demonstrating that Soft-NMS is more effective in maintaining recall and reducing identity switching. Furthermore, replacing the lightweight MobileNetV3 with CSPDarknet53 resulted in a slight improvement in mAP (96.6% vs. 96.3%), but significantly reduced FPS to 17.5 and increased average power consumption to 28.3 W, highlighting the necessity of a lightweight backbone in embedded real-time applications. Finally, replacing the improved version with the original ByteTrack significantly reduced MOTA from 88.3% to 82.4% and IDF1 to 88.6%, indicating that introducing NSA Kalman filtering and GIoU matching is crucial for improving trajectory continuity and identity preservation capabilities in occlusion scenarios. Overall, the modules complement each other in terms of accuracy, stability, and power consumption, jointly supporting the comprehensive advantages of the complete solution in terms of real-time performance and robustness.

G. COMPARISON WITH CURRENT LIGHTWEIGHT STATE-OF-THE-ART MODELS

To demonstrate the advantages of our proposed method compared to similar lightweight real-time tracking models, comparative experiments were conducted on several representative and widely used lightweight detection + tracking combinations under the same dataset partitioning, training strategy, and hardware platform: including YOLOv8n/YOLOv8s (nano and small versions respectively) versus ByteTrack, YOLOv8-MobileNetV3 versus DeepSORT, YOLOX-Nano versus OC-SORT, and EfficientDet-D0 versus ByteTrack. All models were trained for 300 epochs at the same input resolution (640×640) and evaluated on the same test set; if comparable improvements were available in the tracking mechanism, that variant was used; otherwise, the default version of the publicly available implementation was used. Each experiment was repeated 5 times and results are expressed as mean ± standard deviation (n=5); paired t-tests were performed on key metrics (mAP@0.5 , MOTA, IDF1, FPS, average power consumption) to assess significance. The comparison results with the current lightweight SOTA model are shown in Table 5.

Values in Table 5 are mean ± standard deviation from five independent runs (n=5). As shown in Table 5, compared with current mainstream lightweight real-time target tracking models, the proposed YOLOv7-CBAM (MobileNetV3) + improved ByteTrack demonstrates a significant advantage in overall performance. Our method achieves the highest mAP@0.5 (96.3%) in detection accuracy, exceeding

TABLE 4. Ablation Experiment Results

Variants (Ablation Items)	mAP@0.5 (%)	MOTA (%)	IDF1 (%)	FPS	Average power consumption (W)
Complete Solution (MobileNetV3 + CBAM + BiFPN + Soft-NMS + Improved ByteTrack)	96.3	88.3	94.2	32.2	20.7
Remove CBAM (-CBAM)	93.5	84.5	90.7	32.6	20.4
Replace BiFPN with PANet (-BiFPN)	94.5	86.7	92.1	31.0	21.0
Replace Soft-NMS with Hard NMS (-SoftNMS)	95.7	87.5	93.0	33.0	20.6
Replace MobileNetV3 with CSPDarknet53 (+CSPDarknet53)	96.6	87.9	93.5	17.5	28.3
Replace Improved ByteTrack with Original ByteTrack (-ImprovedBT)	95.9	82.4	88.6	33.5	20.5

TABLE 5. Comparison with the current lightweight SOTA model

Model (Detection + Tracking)	mAP@0.5 (%)	MOTA (%)	IDF1 (%)	FPS	Average power consumption (W)
Complete Solution: YOLOv7-CBAM (MobileNetV3) + Improved ByteTrack	96.3	88.3	94.2	32.2	20.7
YOLOv8n + ByteTrack (nano)	91.8	81.5	86.2	38.7	17.9
YOLOv8s + ByteTrack (small)	93.4	84.1	89.0	29.6	21.5
YOLOv8-MobileNetV3 + DeepSORT	86.5	72.4	76.9	24.7	22.4
YOLOX-Nano + OC-SORT	89.2	79.8	83.5	35.1	18.0
EfficientDet-D0 + ByteTrack	88.0	76.0	80.2	21.3	23.5

YOLOv8n + ByteTrack (91.8%) and YOLOv8s + ByteTrack (93.4%) by 4.5 and 2.9 percentage points, respectively. In terms of tracking consistency, its MOTA and IDF1 reach 88.3% and 94.2%, respectively, significantly outperforming YOLOX-Nano + OC-SORT (MOTA 79.8%, IDF1 83.5%) and EfficientDet-D0 + ByteTrack (MOTA 76.0%, IDF1 80.2%), indicating more stable identity preservation capabilities in crowded and occluded scenarios. Although YOLOv8n and YOLOX-Nano achieve frame rates of 38.7 FPS and 35.1 FPS respectively, slightly higher than the 32.2 FPS of our proposed method, their accuracy and identity consistency are significantly reduced, making it difficult to meet the requirements of high-reliability line-of-sight target tracking. From an energy consumption perspective, our proposed method has an average power consumption of 20.7 W, which is lower than YOLOv8s (21.5 W) and YOLOv8-MobileNetV3 (22.4 W), avoiding significant energy consumption while maintaining high accuracy. Overall, these results demonstrate that our proposed method achieves a better balance between accuracy, tracking stability, real-time performance, and power consumption, validating its comprehensive advantages over current lightweight state-of-the-art (SOTA) solutions.

VII. CONCLUSION

This paper constructs an end-to-end real-time tracking framework for passenger gaze targets in urban rail interactive guide interfaces. This framework utilizes MobileNetV3 backbone replacement and CBAM attention recalibration, combined with BiFPN multi-scale weighted fusion and Soft-NMS decoding. MediaPipe and PnP pose estimation are then used to accurately map eye movements to detected interface elements, achieving gaze point localization. Finally, based on the ByteTrack algorithm, the Nonlinear State-Aware (NSA) Kalman filter and Generalized Intersection over Union (GIoU) matching are introduced. Experiments at Terminal 2 of the Beijing Capital Airport Subway Station demonstrate that under moderate occlusion (0.3), detection mAP@0.5 reached 96.3% and tracking MOTP reached 85.7%. The system achieves millisecondlevel performance of 32.2 FPS on an embedded platform. Despite these promising results, the current method still exhibits some errors in gaze mapping under extreme side viewing angles and strong backlighting .

AUTHOR CONTRIBUTIONS

Conceptualization Leilei Zhai, Haishan Xia, JianYe Zhai, Yu qing Qu, Lifeng Li and Fan Yang; methodology Leilei Zhai, Haishan Xia, JianYe Zhai, Yu qing Qu, Lifeng Li and Fan Yang; investigation Leilei Zhai, Haishan Xia, JianYe Zhai, Yu qing Qu, Lifeng Li and Fan Yang; writing-original draft preparation Leilei Zhai, Haishan Xia, JianYe Zhai, Yu qing Qu, Lifeng Li and Fan Yang. All authors have read and agreed to the published version of the manuscript.

CONFLICTS OF INTEREST

The authors declare no conflict of interest.

FUNDING

This work was supported by the National Natural Science Foundation of China(Grant No.52078027).

ACKNOWLEDGEMENTS

Not applicable.

REFERENCES

[1] K. Wang, C. Shen, M. Li, and J. Li, "Research on Users' Willingness to Use the Urban Subway Wayfinding Signage System Based on the DeLone & McLean Model Theory: A Case Study of Wuxi Subway," Systems, vol. 12, no. 12, pp. 529-556, 2024, doi: 10.3390/systems12120529.

[2] X. Li, Y. Zhang, D. He, X. Teng, B. Liu, and Y. Chen, "Passenger flow detection in subway stations based on improved you only look once algorithm," Transportation research record, vol. 2677, no. 9, pp. 397-409, 2023, doi: 10.1177/03611981231159128.

[3] U. Junyan W, U. Xia L I, and I. Yazhuo L, "Application Research on Passenger Flow Real-time Detection of Subway Station Platform Based on Deep," Journal of Jianghan University (Natural Science Edition), vol. 51, no. 5, pp. 67-75, 2023.

[4] W. Zhang, C. Zhu, Y. Qu, G. Liu, and D. H. Lee, "An M-DeepSORT Algorithm for Pedestrian Detection and Tracking Based on Video Images-A Case Study in Ji-nan Subway Station," Promet-Traffic&Transportation, vol. 37, no. 2, pp. 338-360, 2025, doi: 10.7307/ptt.v37i2.635.

[5] M. Sakthi S, M. Ranjitha T, and M. Vidhya Shree, "Real-Time Passenger Counting and Occupancy-Based Announcement System for Metro Trains Using YOLO-Based Deep Learning," Journal of Machine Learning and Signal Processing, vol. 1, no. 1, pp. 1-11, 2025.

[6] N. Li, X. Bai, X. Shen, P. Xin, J. Tian, T. Chai, et al., "Dense pedestrian detection based on GR-YOLO," Sensors, vol. 24, no. 14, pp. 4747-4765, 2024, doi: 10.3390/s24144747.

[7] H. Wang, H. Pei, and J. Zhang, "Detection of locomotive signal lights and pedestrians on railway tracks using improved YOLOv4," IEEE Access, vol. 10, no. 1, pp. 15495-15505, 2022, doi: 10.1109/ACCESS.2022.3148182.

[8] J. Zuo, Y. Wang, and R. Wang, "Method for Acquiring Passenger Standing Positions on Subway Platforms Based on Binocular Vision," IEEE Access, vol. 13, no. 1, pp. 32971-32980, 2025, doi: 10.1109/ACCESS.2025.3543130.

[9] G. Zidani, D. Djarah, A. Benmakhlouf, and L. Khettache, "Optimizing Pedestrian Tracking for Robust Perception With Yolov8 and Deepsort," Applied Computer Science, vol. 20, no. 1, pp. 72-84, 2024, doi: 10.35784/acs-2024-05.

[10] L. Li, F. Gao, S. Ling, Z. Guo, J. Zuo, M. Goodsite, et al., "Would you like to get on the bus? An eye-tracking study based on the stimulus-organism-response framework," Transportation Research Part F: Traffic Psychology and Behaviour, vol. 109, no. 1, pp. 1114-1136, 2025, doi: 10.1016/j.trf.2025.01.014.

[11] Z. Zeng, K. Zhang, and B. Zhang, "Study on the influence of spatial attributes on passengers' path selection at fengtai highspeed railway station based on eye tracking," Buildings, vol. 14, no. 9, pp. 3012-3030, 2024, doi: 10.3390/buildings14093012.

[12] K. Kim J, J. Park, K. Moon Y, and J. Kang S, "Improving gaze tracking in large screens with symmetric gaze angle amplification and optimization technique," IEEE Access, vol. 11, no. 1, pp. 85799-85811, 2023, doi: 10.1109/ACCESS.2023.3282185.

[13] F. Tang, F. Yang, and X. Tian, "Long-distance person detection based on YOLOv7," Electronics, vol. 12, no. 6, pp. 1502-1515, 2023, doi: 10.3390/electronics12061502.

[14] C. Li, Y. Wang, and X. Liu, "An improved YOLOv7 lightweight detection algorithm for obscured pedestrians," Sensors, vol. 23, no. 13, pp. 5912-5925, 2023, doi: 10.3390/s23135912.

[15] J. Hu, Y. Zhou, H. Wang, P. Qiao, and W. Wan, "Research on Deep Learning Detection Model for Pedestrian Objects in Complex Scenes Based on Improved YOLOv7," Sensors, vol. 24, no. 21, pp. 6922-6947, 2024, doi: 10.3390/s24216922.

[16] G. Garcia, A. Velastin S, N. Lastra, H. Ramirez, S. Seriani, and G. Farias, "Train station pedestrian monitoring pilot study using an artificial intelligence approach," Sensors, vol. 24, no. 11, pp. 3377-3397, 2024, doi: 10.3390/s24113377.

[17] S. Ennaama, H. Silkan, A. Bentajer, and A. Tahiri, "Enhanced real-time object detection using YOLOv7 and MobileNetv3," Engineering, Technology & Applied Science Research, vol. 15, no. 1, pp. 19181-19187, 2025, doi: 10.48084/etasr.8777.

[18] C. Liu, Y. Li, J. Gu, Y. Lou, and T. Shen, "Enhancing Pedestrian Target Recognition in Open Community Multi-Scene Spaces Using the Yolo-Stp Network," ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. 10, no. 1, pp. 319-328, 2023, doi: 10.5194/isprs-annals-X-1-W1-2023-319-2023.

[19] Y. Hao and C. Geng, "Improved pedestrian vehicle detection for small objects based on attention mechanism," Int. J. Adv. Netw. Monit. Control, vol. 9, no. 1, pp. 80-89, 2024, doi: 10.2478/ijanmc-2024-0030.

[20] A. Assefa A, W. Tian, N. Acheampong K, M. U. Aftab, and M. Ahmad, "Small-Scale and Occluded Pedestrian Detection Using Multi Mapping Feature Extraction Function and Modified Soft-NMS," Computational intelligence and neuroscience, vol. 2022, no. 1, pp. 1-11, 2022, doi: 10.1155/2022/9325803.

[21] H. Khaleel A, H. Abbas T, and W. Sami Ibrahim A, "Best low-cost methods for real-time detection of the eye and gaze tracking," i-com, vol. 23, no. 1, pp. 79-94, 2024, doi: 10.1515/icom-2023-0026.

[22] C. Leblond-Menard and S. Achiche, "Non-intrusive real time eye tracking using facial alignment for assistive technologies," IEEE Transactions on Neural Systems and Rehabilitation Engineering, vol. 31, no. 1, pp. 954-961, 2023, doi: 10.1109/ TNSRE.2023.3236886.

[23] S. Zhuang, Z. Zhao, L. Cao, D. Wang, C. Fu, and K. Du, "A robust and fast method to the perspective-n-point problem for camera pose estimation," IEEE Sensors Journal, vol. 23, no. 11, pp. 11892-11906, 2023, doi: 10.1109/JSEN.2023.3266392.

[24] P. Roch, B. Shahbaz Nejad, M. Handte, and P. J. Marron, "Axes-aligned non-linear optimized PnP algorithm," Machine Vision and Applications, vol. 35, no. 6, pp. 137-150, 2024, doi: 10.1007/s00138-024-01618-z.

[25] L. You, Y. Chen, C. Xiao, C. Sun, and R. Li, "Multi-object vehicle detection and tracking algorithm based on improved YOLOv8 and ByteTrack," Electronics, vol. 13, no. 15, pp. 3033-3048, 2024, doi: 10.3390/electronics13153033.

[26] G. Jang J and W. Kwon Y, "Trajectory Similarity-Based Traffic Flow Analysis Using YOLO+ ByteTrack," Journal of Multimedia Information System, vol. 12, no. 1, pp. 27-40, 2025, doi: 10.33851/JMIS.2025.12.1.27.

[27] Q. Wang, G. Ye, Q. Chen, S. Zhang, and F. Wang, "A UAV perspective based lightweight target detection and tracking algorithm for intelligent transportation," Complex & Intelligent Systems, vol. 11, no. 1, pp. 73-90, 2025, doi: 10.1007/ s40747-024-01687-7.

[28] N. Zelin R E, P. Lan, W. Chao, L. Jiaheng, and Z. Fangyan, "Low-light pedestrian detection and tracking algorithm based on autoencoder structure and improved Bytetrack," Journal of Applied Optics, vol. 45, no. 3, pp. 616-629, 2024, doi: 10.5768/JAO202445.0302001.