Received: 17 November 2023 | Revised: 13 March 2024 | Accepted: 9 April 2024 | Published online: 14 April 2024

RESEARCH ARTICLE

Artificial Intelligence and Applications
2025, Vol. 3(4) 459–472
DOI: 10.47852/bonviewAIA42022098

BON VIEW PUBLISHING

Side Collision Detection Model for Visually Impaired Using Monocular Object-Specific Distance Estimation and Multimodal Real-World Location Calculation

Wenqing Song[1,*], Yumeng Sun[1], Qixuan Huang[1] and Junyang Cheok[1]

[1]*Faculty of Computer Science and Information Technology, University of Malaya, Malaysia*

Abstract: Targeting the potential risk of side vehicle collisions when the visually impaired crosses roads, this study proposed a side collision detection model, including monocular distance estimation, multimodal real-world location estimation, future location prediction, and collision warning strategies tailored for visually impaired pedestrians. The proposed model employs YOLOv8 and DeepSort for vehicle detection and tracking, utilizing shallow neural networks for distance estimation based on image information and vehicle position data. Predicted vehicle distances are combined with magnetic field sensor and GPS data to compute and store real-world vehicle locations, and these location data will be used for linear regression to forecast future locations. A warning strategy is then implemented to alert users. Experimental validation shows that the monocular distance estimation network has an Absolute Relative Error of 0.043 and an ALE (Average Localization Error) of 1.249 m. In real-world location estimation, the view angle ALE is 0.019, and the location ALE is 1.778 m. Regarding location prediction, the accuracy in distinguishing stationary and moving vehicles reaches 0.962, and the predicted curve, based on ground truth and predicted locations, exhibits good alignment. The proposed warning strategy, evaluated on Kitti Tracking Dataset and a self-created dataset, accurately detects the majority of potential collision risks.

Keywords: side collision detection, the visually impaired, monocular distance estimation, real-world location estimation

1. Introduction

139 million people were blind or visually impaired globally in 2019; according to the Global Burden of Disease research [1], 336 million people globally, including 206 million with moderate visual impairment and 130 million with severe visual impairment or blindness, have moderate to severe visual impairment. Their everyday tasks are significantly hampered by their blindness, especially mobility and navigation.

There has been substantial progress in navigation systems for the blind. For instance, electrical vision aids, developed in the 1960s, employ cameras and electrical signal processing to magnify or enhance distant sights, enabling the visually handicapped to more easily recognize objects and settings. The ability for the blind to hear text and information on electronic devices through spoken output was made possible in the late 1970s thanks to screen readers and voice synthesis technology. Sensing technology has been used to improve blind people's placement and navigation since the 1990s. These systems help users avoid obstacles and reach their goal via sound, vibration, or tactile cues with feedback. But research on side collision warning and prediction methods is still lacking. Real-time assistance systems made up 80% of the video cameras utilized, according to a review by Mashiata et al. [2], with RGB-D and monocular cameras each contributing 40%. There are now two types of navigation systems for visually impaired people: interior systems and outdoor systems. These systems focus mainly on obstacle detection, GPS-based path guiding, monocular video, RGB-D cameras, and sonar sensors, with instructions transmitted via voice, vibrations, or stimuli.

These systems typically concentrate on object identification and recognition in front of the user, even though they span a wide range of scenarios and purposes, such as navigation in shopping malls or airports [3]. Unfortunately, they fail to detect and acknowledge moving things that are on the side of user. This oversight creates a severe risk for visually challenged people who attempt to cross roads without signal lights since they run the chance of being hit by cars coming from both directions.

This paper proposes a side collision detection model for the visually impaired using mobile devices. The paper is structured as follows: Section 2 summarizes the related works, whereas Section 3 describes the proposed side collision detection model. Experiments are detailed in Section 4, followed by conclusions.

2. Related Work

The field of visually impaired outdoor assistance model has seen extensively research. Ramadhan [4] proposed a wearable system composed of a micro-controller board, a solar panel,

*Corresponding author: Wenqing Song, Faculty of Computer Science and Information Technology, University of Malaya, Malaysia. Email: S2199451@siswa.um.edu.my

© The Author(s) 2024. Published by BON VIEW PUBLISHING PTE. LTD. This is an open access article under the CC BY License (https://creativecommons.org/licenses/by/4.0/).

some sensors, mobile communication, and GPS modules. The proposed wearable system employed a set of sensors to trace the path and to warn of obstacles in front of visually impaired users. Elmannai and Elleithy [5] integrated sensor fusion, computer vision, and fuzzy logic techniques to provide accurate multi-object detection for collision avoidance. Croce et al. [6] presented an indoor and outdoor navigation system in which both a camera and inertial sensors were integrated into a smartphone. Li et al. [7] described a wearable application that uses an RGB-D camera and an inertial measurement unit (IMU) to identify objects in real-time and create collision-free routes to assist visually impaired persons in swiftly gaining situational awareness and walking safely. Tian et al. [8] combined the depth camera and object recognition, realizing the detection and early warning of obstacles. And they utilize neural network training to identify traffic light signals and sidewalks, which gives users more road condition information. Ahmed et al. [9] and Meliones et al. [10] proposed the obstacle detection model using ultrasonic sensor. They determine the range and distance of obstacles by processing the received data from two sensors. See et al. [11] proposed an obstacle detection model exclusively on a smartphone.

Based on the position of the camera relative to the vehicle and the road, as well as the vehicle's velocity, each vehicle can have a set of N measurements. The initial decision to make is which distance between which measurements to use for obtaining velocity values. Most studies suggest using consecutive measurements (at frames t and $t+1$) to compute the velocity values for each vehicle [12]. In cases where non-consecutive images are used, several techniques have been proposed in existing research. These include using fixed distances or measurement areas [13], using predetermined frame intervals [14], or fixed time intervals [15] for measurements. More recently, data-driven approaches based on end-to-end deep learning methods can be found, where velocity detection is modeled as a regression problem. For example, in the work by Lee et al. [16], two different neural networks were trained using animated images (overhead view of two-lane road sections) from simulators and synthetic images generated through CycleGAN. With standard convolutional architectures, the output fully connected layer represents the average speed of the road. In Dong et al. [17], vehicle speed estimation was treated as a three-dimensional convolutional network for video action recognition.

Today, there is extensive research on 3D object detection and image depth estimation, but there is still relatively limited research on object-specific distance estimation. So, for object distance estimation, it is typically achieved by training a monocular depth estimation network with a dataset to obtain distance results [18, 19], extracting predicted depth information from 3D object detection [20–22], or deriving absolute depth information from the results of stereo depth estimation [23].

Recently, there are some researches for object-specific distance estimation. Ali and Hussein [24] and Bertoni et al. [25] proposed object distance regression model using height and pose detection. After that, Zhu and Fang [26] proposed a directly distance regression network based on the object feature map. They utilize ResNet or VggNet as backbone to extract the feature of image, and then, RoI pooling is used to get the object feature. Finally, the distance is output after several FC layers. There are limitations for their model's accuracy, but work of Zhu and Fang [26] inspired Zhang et al. [27] to develop another object-specific distance estimation model based on Mask RCNN. They extract the object feature from backbone according to RPN net, adding another distance ahead the same as box regression head to get the distance. The outcome indicates their model has excellent accuracy.

In conclusion, outdoor assistive models for the blind typically use GPS for navigation, combined with depth cameras, radar sensors, and image recognition for obstacle detection and warnings. Object detection can identify objects such as traffic lights and crosswalks, providing better road information. However, these models do not take into account the risk that a blind person may face from a side vehicle collision when crossing the road without a signal light. Furthermore, for our proposed model, the maximum estimated depth of the depth camera may not meet our requirements.

For vehicle speed estimation, the first is to use a fixed camera to estimate the moving distance of the vehicle between frames through object detection or key point detection to obtain the average speed. The second is to use a moving camera to obtain the distance of the vehicle through sensors or distance estimation, and then obtain the relative speed of the vehicle through calculation. First of all, in outdoor assistive equipment for the blind, the blind need to move. Secondly, the movement of vehicles is relatively stable, while people's body angles will change during the movement, so using relative speed calculations will produce large errors.

So, we proposed a side collision detection model. The proposed model (1) performs depth estimation using a shallow depth estimation network with detection parameters and image information, (2) calculates the real-world location of the object through GPS and magnetic field sensor, (3) estimates the object's future location through continuous measurements of its current location, and (4) proposes a collision detection strategy that can be applied to visually impaired pedestrians.

3. Research Methodology

3.1. Model design

Figure 1 illustrates the model diagram. While the model is running, the system will continuously acquire real-time data from the camera, magnetic sensor and GPS.

Initially, the model detects vehicles in the current frame, extracting their features and comparing them with existing features to determine whether they are new vehicles. Subsequently, the model takes the position information of each vehicle in the frame, inputs it into the deep regression network, and combines it with field of view, GPS and magnetic sensor data to calculate the vehicle's real-world location. This calculated location is then put to the corresponding vehicle location queue.

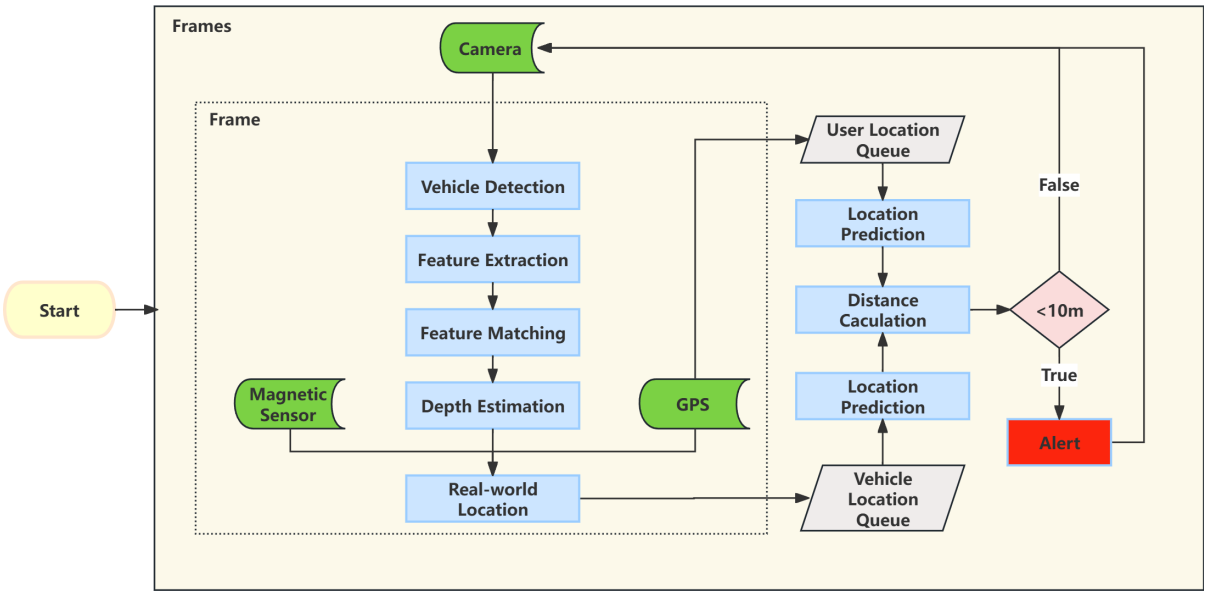
With the vehicle location queue and user location queue from the GPS, the model predicts the locations of both users and vehicles for the next few seconds. During the prediction, if distance between user and vehicle is less than 10 meters, then the model will trigger an alert.

3.2. Vehicle detection and tracking

This module is designed for the detection and tracking of vehicles, enabling further location estimation and prediction. Our model utilizes YOLOv8 [28] and DeepSort [29] to accomplish this function.

In each frame, the model employs the YOLOv8n model to detect vehicles and extract their parameters. Subsequently, DeepSort is utilized to identify landmarks and extract distinctive features for each vehicle. In the subsequent frames, the model conducts feature matching with the existing features. If success, then the car will be recognized as the existing one else a new one. The camera parameters and label information will be utilized in the location estimation and prediction section.

Figure 1
Model diagram



3.3. Location estimation and prediction

3.3.1. Distance estimation

According to Zhu and Fang [26], we propose a similar depth regression network with the object position information and the image data. We can directly regress the object depth or regress the height. With the pixel height h_w and the real-world height h and the focal length in y axis, we can calculate the depth by the following formula, which is also the a priori formula of many 3D detection models.

$$d = \frac{h_w f_y}{h} \tag{1}$$

In addition to using bounding boxes, we also leverage the vehicle's image position as a means to infer depth information. As is demonstrated in Figure 2, We consider several image parameters as input, including pixel width, pixel height, the pixel location along the y axis from the bottom of the image to the bottom of the vehicle, and the horizontal pixel distance from the image's center to the center of the vehicle. Then with a FC layer, the 4 parameters are reshaped into 16 hidden neurons.

For image data, we directly utilize 8×8 ROI_Align to extract the vehicle feature after normalization. Then, we flatten the feature and employed FC layer to reshape the flattened feature into a 1×16 tensor.

Next, we concatenate the bounding box tensor with the image tensor. Subsequently, we employ additional FC layers to regress the tensor into the final depth output. ReLU functions are employed as activation functions throughout the network. The depth estimation methods have a significant margin of error in long-distance predictions. So, we utilize $L1$ loss function to reduce the impact of outliers.

$$loss_{l1}(GT, PR) = \frac{\sum_{i=1}^{n} |GT_i - PR_i|}{n} \tag{2}$$

After calculating the view angle, the distance can be calculated in the formula below. View angle stands for the horizontal view angle from the camera forward direction to the vehicle. x stands for the horizontal middle coordinate of the vehicle. Width stands for the horizontal pixels of the image. f_x stands for the horizontal focal length of the camera in pixel.

$$View\ Angle = \frac{x - 0.5 \times width}{f_x} \tag{3}$$

$$distance = \frac{depth}{\cos(|View\ Angle|)} \tag{4}$$

3.3.2. Location calculation

In this section, we compute the real-world location of the vehicle for subsequent prediction. The reason we utilize this method is that users may perform continuous turns while crossing the road or standing aside. As shown in Figure 3, when the user moves from G_1 to G_2, there is a change in the orientation of the camera (α to β).

In such case, relying solely on relative position for prediction can introduce significant errors. With real-world location, the proposed model is capable to predict the vehicle location in real-world. To address this, we calculate the distance using depth information and view angles obtained from the depth estimation section. Then, with additional data from GPS, magnetic sensors, distance, and view angles, we compute the vehicle's real-world location by GeographicLib, as is shown in the formula below.

$(\tilde{lat}_i, \tilde{lon}_i)$: The real-world location of user from GPS.

$\hat{angle}_i + \hat{mag}_i, \hat{distance}_i$: The estimated view angle and distance from user to the car and the magnetic sensor value. Angle with north as 0 degrees, clockwise as positive.

$(\hat{lat}_i, \hat{lon}_i)$: The estimated location of the vehicle.

$$(\hat{lat}_i, \hat{lon}_i) = geod.Direct(\tilde{lat}_i, \tilde{lon}_i, \hat{angle}_i + \hat{mag}_i, \hat{distance}_i) \tag{5}$$

3.3.3. Location prediction

For data storage structure, we created individual timing queues for each object to store their location data. The maximum queue size

Figure 2
Flowchart for location estimation and prediction

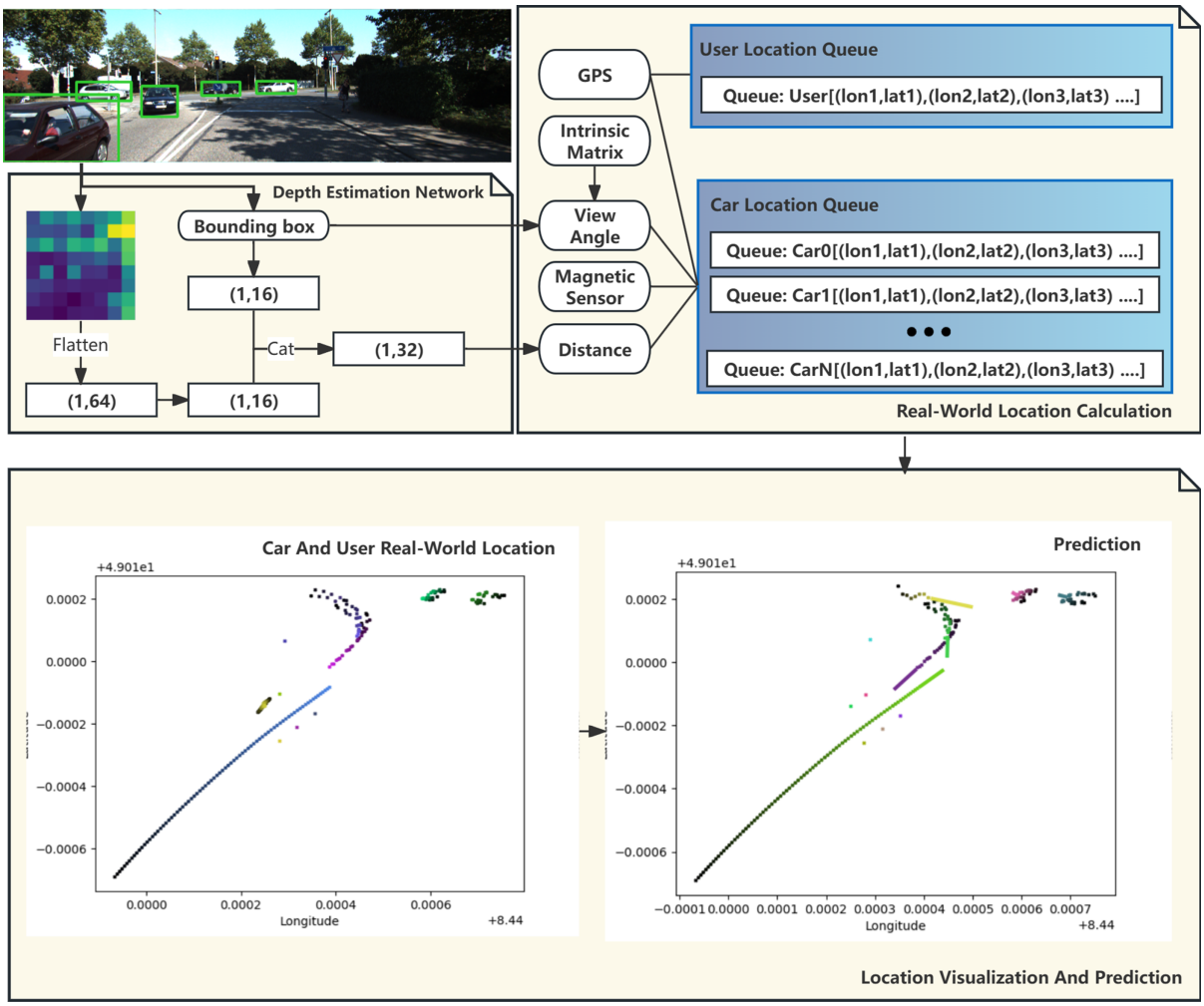
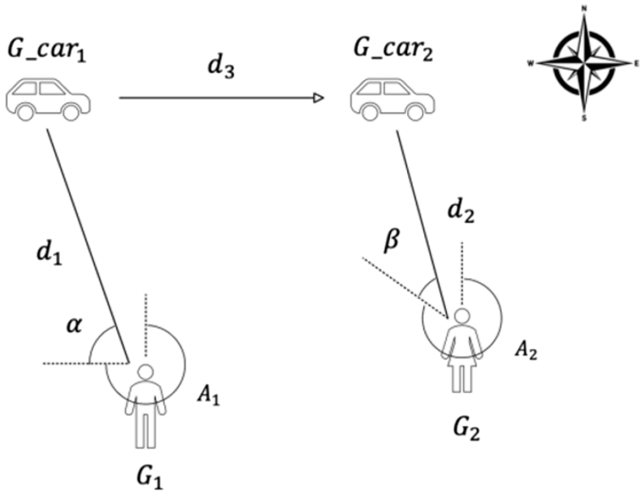


Figure 3
Schematic diagram of model application scenario



is set as 160 frames. Upon the arrival of new frame data, a new queue is created for new detected objects, and relevant data will be inserted into the existing queues for objects detected in previous frames. For objects not detected in a particular frame, we insert invalid values into their respective queues. The undetected objects in this frame are still retained to mitigate the impact of occlusion. To optimize computational resources, we implement a sleep parameter to count the consecutive number of frames in which an object remains undetected. When this parameter reaches a threshold of 40, we destroy the corresponding queue.

The next step involves distinguishing between stationary and moving vehicles. The invalid data insertion is to synchronize the time, so we filter those invalid data at first. In monocular estimation, the regression distance has error, so does the real-world location. Consequently, the vehicles we detect will keep moving, even if it is still. To distinguish between stationary and moving objects, we calculate the standard deviation of the entire queue data and set a threshold 1e-10. If the standard deviation falls below 1e-10, we consider it as the stationary object and filter out these objects. The remaining vehicle queue data will be used for location prediction.

$$s = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2} \tag{6}$$

$$s(lat) < 1e-10 \, and \, s(lon) < 1e-10 \Rightarrow stationary \tag{7}$$

For location prediction, we employ the preceding 80 frames of valid locations to forecast the location of objects for the subsequent 100 frames, which is shown in Location Visualization and Prediction part of Figure 2. In an effort to reduce the influence of errors, we utilize the Linear Regression function from Scikit-learn for location prediction. We treat time t as the independent variable and latitude and longitude as the dependent variables, generating two linear functions that relate latitude and longitude to time.

$$lat_t = k_1 t + b_1 \quad lon_t = k_2 t + b_2 \tag{8}$$

$$location_t = (lat_t, lon_t), t \in [0, 99] \tag{9}$$

3.4. Collision detection

For the stationary vehicle, we utilize the same strategy of obstacle detection like the work by See et al. [11]. For the moving vehicle, we implement the strategy of "Predicting 5 s, would it be a vehicle less than 10 meters."

Building upon the previous section, where we obtained the predicted locations of both the user and the vehicle, our strategy involves calculating the distance between each predicted location of the user and the vehicle. We then calculate the minimum distance among these calculations. If the minimum distance is less than 10 meters, then the model triggers an alert. The formula for calculating distance at time t is provided below. The Geod.Inverse() function calculates the distance in meters between two real-world locations, where lat_u and lon_u represent the real-world location of the user, and lat_c and lon_c denote the real-world location of the car.

$$distance_t = geod.Inverse(lat_u, lon_u, lat_c, lon_c) \tag{10}$$

$$distList = [distance_0, distance_1, \ldots;, distance_{99}] \tag{11}$$

$$min(distList) < 10 \Rightarrow Alert \tag{12}$$

The advantage of this approach is that it triggers an alert even in scenarios where the user can safely cross the road but a vehicle passes by, or when the user is stationary.

4. Experimental Results

4.1. Evaluation metrics

The objective of our depth estimation is to accurately predict object depth values as close as the ground truth values. We adopt evaluation metrics from Zhang et al. [27], which includes absolute linear error (ALE), absolute relative error (Abs Rel), root mean squared error (RMSE) and relative error threshold (δ < threshold).

For assessing location estimation precision, we employ ALE. For view angle error evaluation, we also use the ALE. The calculation of distance error is performed using GeographicLib with equatorial radius set as 6378388 and flatten set as 1/297. The location ALE can be determined using the following formula.

(lat_i, lon_i): The GT location of the vehicle at time i.

$(\hat{lat}_i, \hat{lon}_i)$: The PR location of the vehicle at time i.

$$ALE_{location} = \frac{1}{n}\sum_{i=1}^{n}|geod.Inverse(lat_i, lon_i, \hat{lat}_i, \hat{lon}_i)| \tag{13}$$

For stationary vehicle determination evaluation, we calculate the success rate by dividing the number of successful decisions by the number of GT count. Since this model is primarily intended for use on straight road sections, we employ a linear equation to predict vehicle locations. Our evaluation does not focus on the effectiveness of linear regression function itself but rather evaluate the degree of fit of the path function in comparison to the ground truth location and predicted location. We employ the R-square metric as our evaluation method.

To evaluate on the collision detection model, we set the prediction time as 5 s and the alert distance threshold as 10 meters. And to reduce the impact of outliers, we filter out vehicles with coordinate sequence length less than 5. After that, we classify the vehicle status into two class: Safety and Risk. Next, we separately count the sample data for safety and risk under the PR and GT conditions. We evaluate the performance of collision detection through precision, recall, $F1$-score, ROC (receiver operating characteristic), and AUC (area under the curve).

4.2. Implementation details

For depth estimation training, we use the Adam optimizer with a learning rate of 0.01 with 0.9 decay every 5 iterations. Batch size is set as 32. Training was performed for 500 iterations. The experiments are performed on Kaggle platform with GPU P100, torch 2.0.0, and cuda 11.8.

4.3. Kitti object dataset

The Kitti object dataset [30] is utilized for depth regression network training and validation. Given the sensitivity of the proposed model to detect bounding box parameters, we filtered out partially truncated cars. We divided the total of 7481 images into two subsets, a training dataset comprising 4000 images with 13078 cars and a validation dataset containing 3481 images with 11597 cars. Figure 4 illustrates the distribution of cars in meters.

Figure 4
Car distribution across distance in train and valid dataset

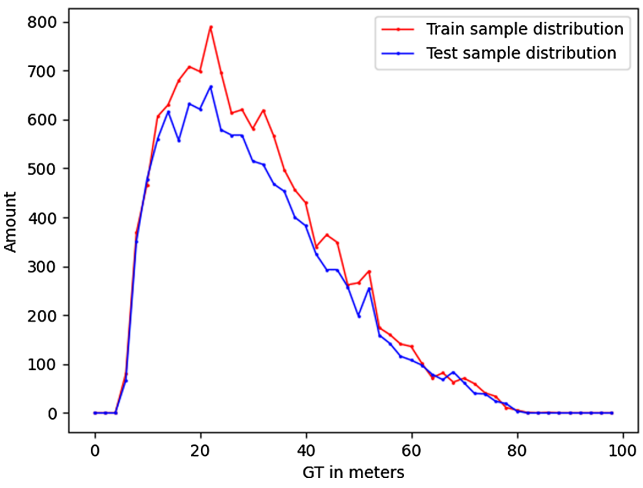
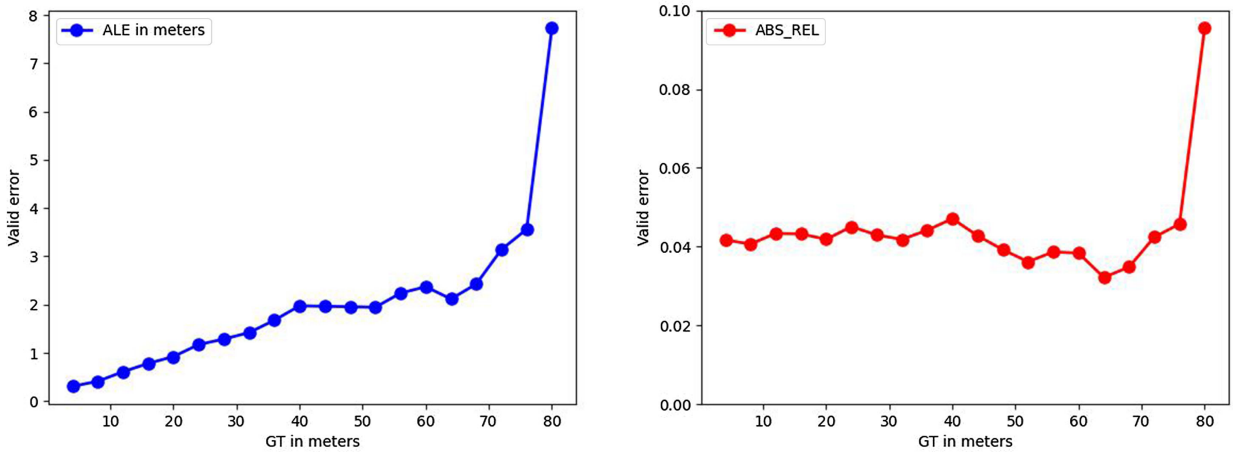


Figure 5
The ALE and Abs Rel of the proposed depth estimation network



We can see that the vehicle distance is mainly concentrated within the range of 5 to 60 meters.

After 150 iterations, the train loss tends to be stable. Figure 5 presents Valid ALE and Abs Rel in meters. Additionally, to compare with other models, we trained the network separately for pedestrians and bicycles. Table 1 illustrates our network's performance. According to the ALE in meters image, we observe an increase in prediction error as the ground truth depth becomes greater. The error increases sharply between 76 and 80 meters, primarily attributed to the scarcity of vehicle samples at the corresponding distance. In the Abs Rel graph, the error rate fluctuates around 0.4 and has a relatively excellent accuracy rate at the middle distance of 40–70 meters, which is beneficial to our subsequent location estimation of distant vehicles.

We follow Zhang et al. [27], dividing distance detection into four types: (A) distance estimation with depth map, (B) GT bounding boxes based distance estimation, (C) 3D object detection, and (D) distance estimation combined with 2d object detection.

The comparison results are presented in Table 2. The evaluation data of M3D-RPN, D4LCN, and SMOKE are from Zhang et al. [27]. The evaluation data of the work by Lee et al. [19] and DEOM-Car are targeted to car exclusively. For the work by Lee et al. [19], there are no ALE data in the comparison table, and the best ALE value in the ablation study is utilized as the ALE. The baseline model is trained with the bounding box parameter only. In this table, 3D detection methods exhibit relatively strong performance, while 2D detection approaches, especially self-supervised depth estimation methods, show limitations in estimating object distances. In addition, the monocular distance estimation method based on Mask RCNN performs well across all evaluation parameters.

Table 1
Performance of the proposed model of different object classes on Kitti object dataset

	Higher is better		Lower is better		
Object class	δ<1.05	δ<1.25	ALE	Abs Rel	RMSE
Car	0.673	**0.994**	1.301	**0.043**	1.995
Pedestrian	**0.678**	0.994	**0.857**	0.044	**1.342**
Cyclist	0.612	0.994	1.077	0.048	1.495
Total	0.672	0.994	1.249	0.043	1.924

Table 2
Performance of different distance estimation models on Kitti object dataset

	Higher is better			Lower is better	
Approach	δ<1.05	δ<1.25	ALE	Abs Rel	RMSE
(A) Liang et al. [18]	–	0.899	–	0.101	–
(A) Lee et al. [19]	–	0.982	1.166	0.047	2.091
(B) DistFormer [31]	–	0.937	–	0.104	2.950
(B) Zhu and Fang [26]	–	0.629	–	0.251	6.870
(B) DEOM-CAR [27]	–	0.992	–	0.046	**1.645**
(B) Baseline	0.576	0.991	1.629	0.052	2.488
(B) Ours	**0.672**	**0.994**	1.249	**0.043**	1.924
(C) M3D-RPN [21]	0.532	–	1.314	0.060	2.050
(C) D4LCN [20]	0.606	–	**1.162**	0.052	1.876
(C) SMOKE [22]	0.561	–	1.412	0.056	2.151
(C) Mauri et al. [32]	–	0.941	–	0.096	2.960
(C) Jing et al. [33]	–	0.976	–	0.069	2.503
(D) Mask RCNN [27]	0.610	–	1.165	0.051	1.943
(D) Vajgl et al. [34]	–	–	2.570	0.110	–

Table 3
Performance of different distance estimation model under different object classes on Kitti object dataset

Approach	Object class	δ<1.25	Abs Rel	RMSE
DistFormer [31]	Car	0.943	0.099	2.11
Zhu and Fang [26]		0.848	0.161	3.580
Mauri et al. [32]		0.941	0.096	3.050
DEOM [27]		0.992	0.046	**1.645**
Ours		**0.994**	**0.043**	1.995
DistFormer [31]	Pedestrian	0.982	0.057	1.26
Zhu and Fang [26]		0.747	0.183	3.439
Mauri et al. [32]		0.935	0.098	2.010
DEOM [27]		0.991	0.049	2.043
Ours		**0.994**	**0.044**	**1.342**
DistFormer [31]	Cyclist	0.956	0.080	3.09
Zhu and Fang [26]		0.768	0.188	4.891
Mauri et al. [32]		0.940	0.098	3.570
DEOM [27]		**0.995**	**0.046**	**1.141**
Ours		0.994	0.048	1.495

The proposed model outperforms other models in terms of relative error threshold and Abs Rel, but does not reach the optimum in ALE and RMSE, mainly because the error of the detection frame estimation method becomes larger with increasing distance. In the meanwhile, occasional estimation outliers contribute to the higher RMSE values observed with our method.

Table 3 presents the performance of various models in distance estimation for different object categories. Zhu, DEOM, DistFormer and our methods are all based on ground truth bounding box information. The proposed model and DEOM demonstrate the best performance in estimating distances for pedestrians and cyclists, respectively. When it comes to vehicle distance estimation, our method exhibits a lower error rate. However, DEOM attains a lower RMSE, the reason for which may be comparatively less robust in long-distance vehicle distance estimation of our model. Images in Figure 6 are some test outputs of the proposed model.

4.4. Kitti tracking dataset

The Kitti tracking dataset [30] is employed for both location estimation and location prediction evaluations. In this section, we focus on the estimation and prediction of car locations.

For the location estimation evaluation, we looped 20 videos with a total of 20,683 location predictions. The outcomes are presented in

Figure 6
Distance estimation test outputs on Kitti object dataset

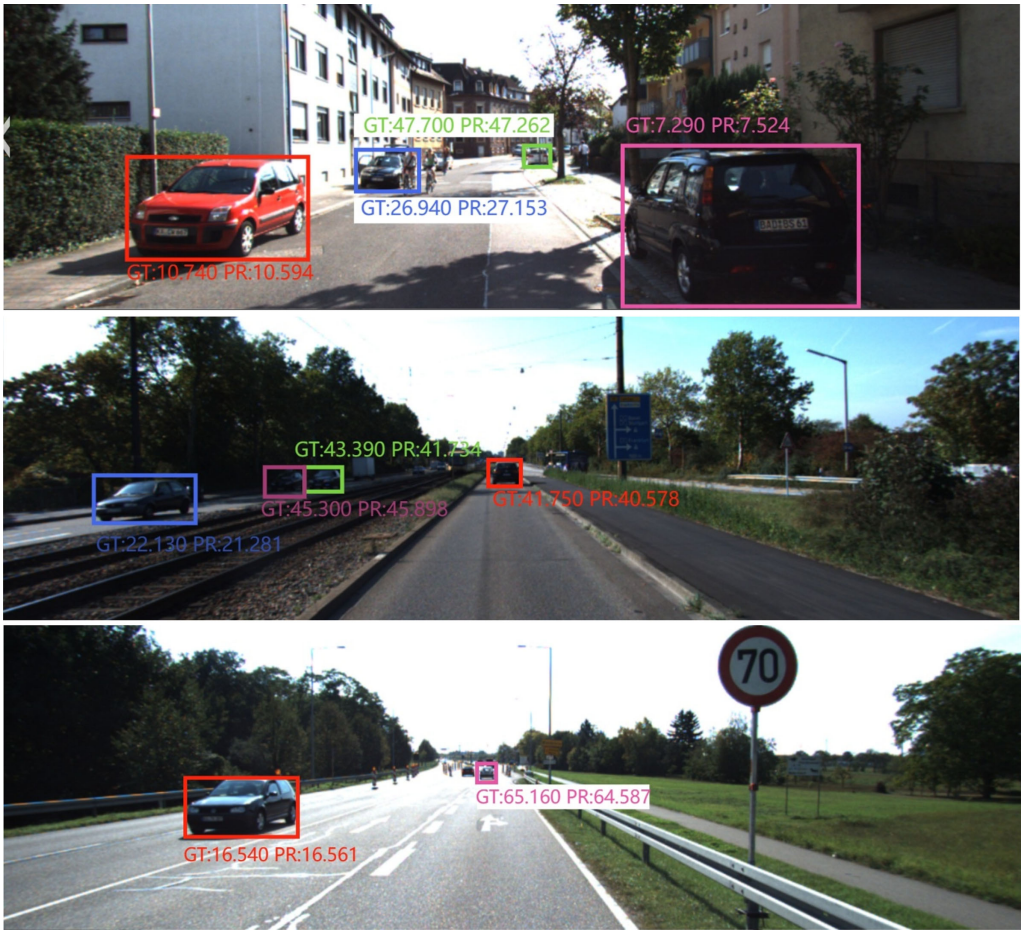


Figure 7
ALE in meters of the view angle and location on Kitti tracking dataset

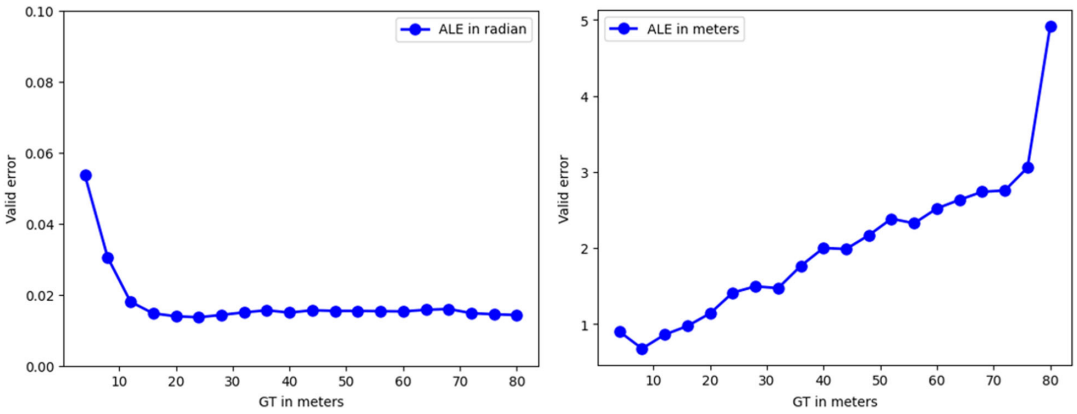


Table 4, and Figure 7 illustrates the average local errors. With the ground truth view angle, the location ALE in meters is 1.344 m, close to depth estimation of the vehicle in our model.

Table 4
Location estimation evaluation on Kitti tracking dataset

Approach	View angle ALE (radian)	Location ALE (meter)
Ours (GT view angle)	0	1.344
Ours	0.019	1.778

The figure reveals that, because of the truncation of the vehicle, the error in field of view angle estimation and location estimation is relatively prominent, particularly when the distance is very close. Moreover, at longer distances, the location estimation error is better than depth estimation error of our model trained with the Kitti Object dataset. This enhancement can be attributed to the fact that vehicles at long distances in the tracking dataset primarily travel in straight lines, leading to increased accuracy. In terms of view angle error, the location estimation error experiences a modest increase of approximately 0.4 meters, which remains within an acceptable range.

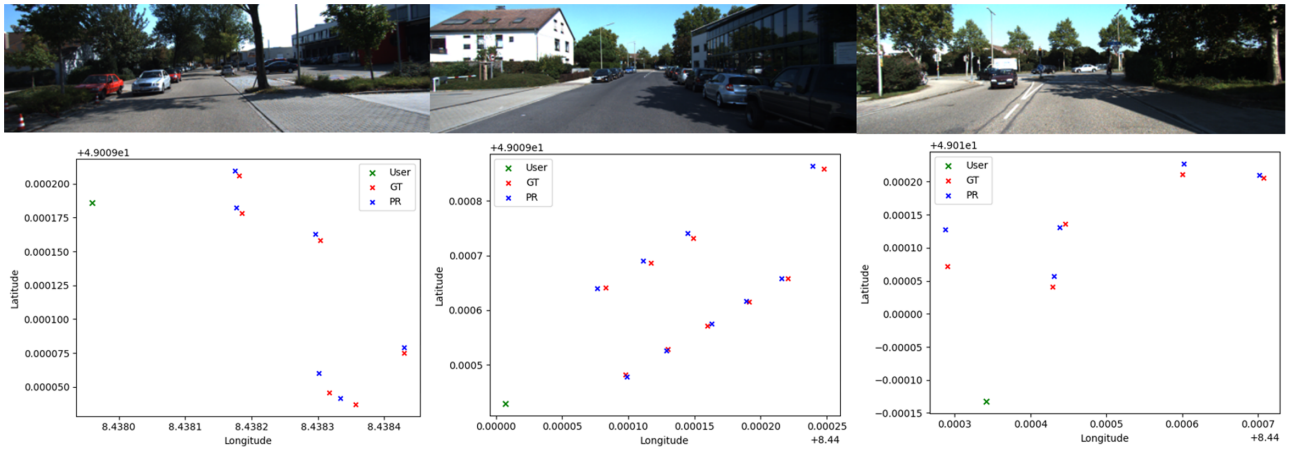
Figure 8 provides some test outputs of location prediction. In the outputs figure, X axis and Y axis represent longitude and latitude, respectively. The color of User location, GT vehicle location, and PR vehicle location are green, red, and blue, respectively.

For the evaluation of location prediction, we only utilized video 1 from the Kitti tracking dataset, consisting of 447 frames. During the experiment, we detected a total of 5870 stationary cars, of which 5649 cars were successfully identified. Additionally, there were 116 regressions linear. The results are presented in Table 5. The success rate is 0.962. This success rate

Table 5
Evaluation outcomes of the location prediction on Kitti tracking dataset, video 1

Approach	Stationary car determination success rate	R-square score
Ours	0.962	Latitude: 0.747, Longitude: 0.766

Figure 8
Real-world location estimation test outputs on Kitti tracking dataset, video 1



is subject to increase as the standard deviation threshold for classifying stationary cars is raised. However, raising this threshold may lead to temporary wrong classification of moving vehicles as stationary. The accuracy of the predicted longitude and latitude slightly deviate from the ground truth values. Test outputs are shown in Figure 9.

To evaluate the collision detection module, according to the type of roads, we categorize the 20 videos into three groups: streets, roads, and highways. We take two videos of each category respectively for experiment. Table 6 illustrates the results. Figure 10 shows the ROC curve and the AUC score for each scenario. In the prediction of safety categories, the evaluation parameters of the proposed model consistently maintain very high standards across all scenarios. More importantly, for the RISK category predictions, the model maintains recall values above 0.9 in all scenarios except for the highway 1 scenario. This indicates that the proposed model can accurately predict collision risks for the majority of cases and issue alerts. In the case of the highway 1

Figure 9

Location prediction test output with current frame, GT location, GT location prediction, PR location, and PR location prediction

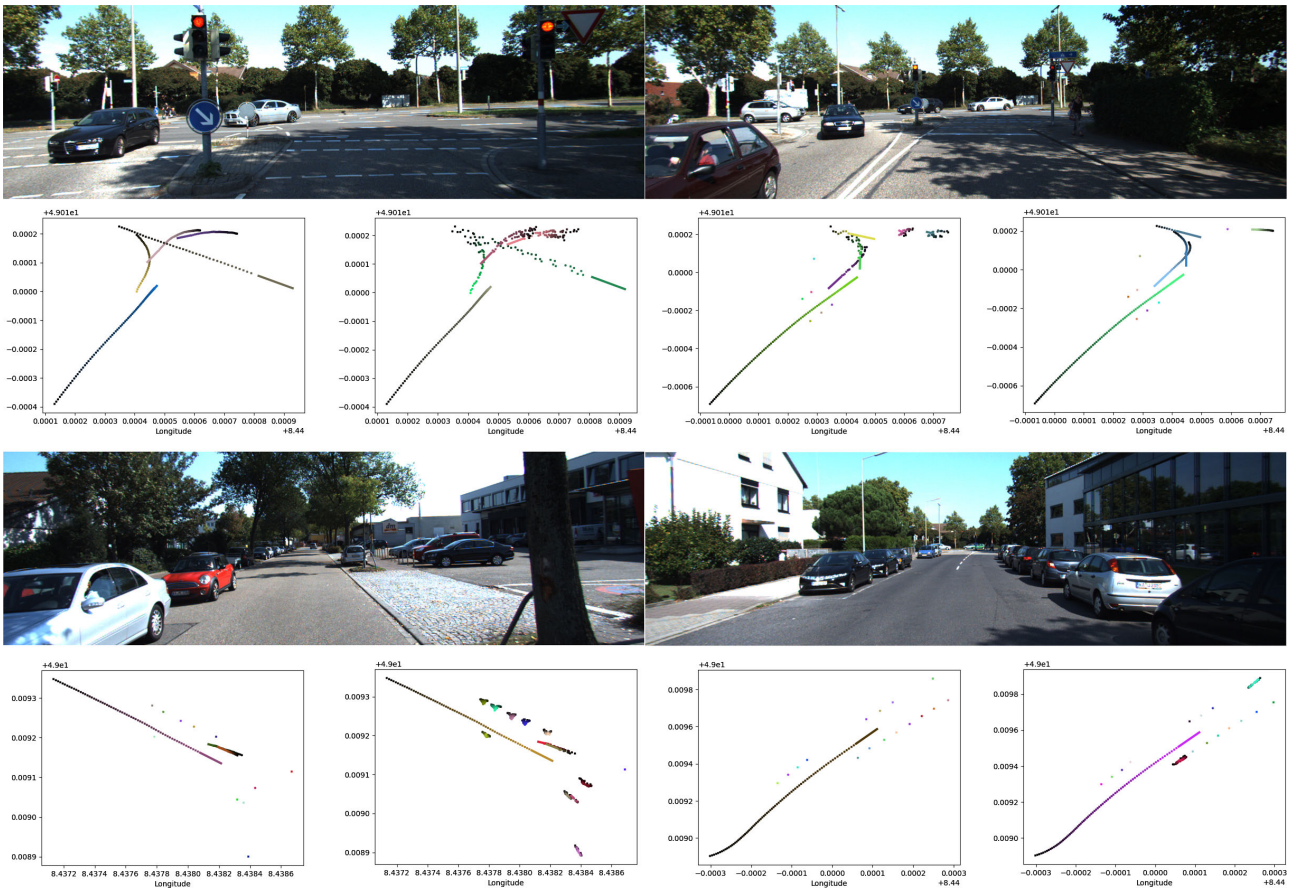
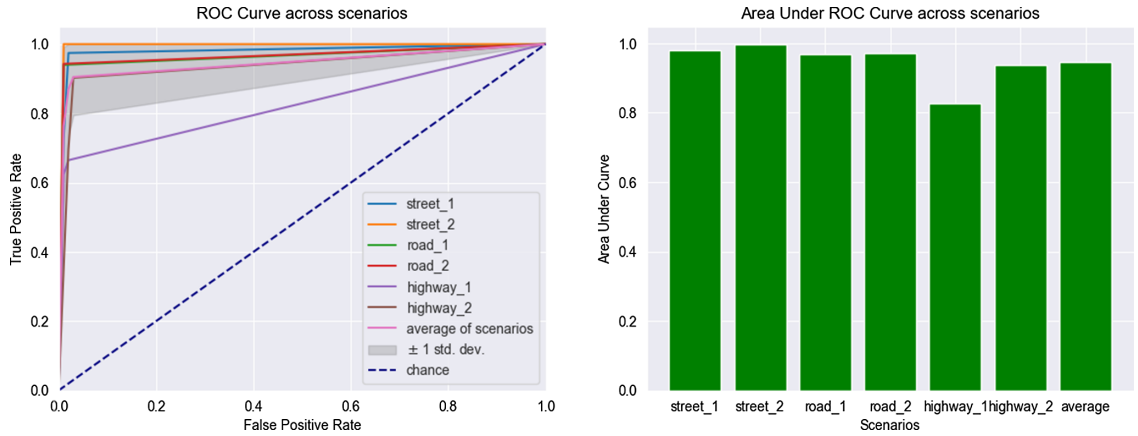


Table 6

Evaluation outcomes of collision detection on Kitti tracking dataset

Scenario	Status	Precision	Recall	F1-score	Support	AUC
Street 1 (Video 1)	Safety	1.00	0.99	0.99	6025	
	Risk	0.70	0.97	0.81	194	0.98
Street 2 (Video 9)	Safety	1.00	1.00	1.00	6032	
	Risk	0.63	1.00	0.77	41	1.00
Road 1 (Video 5)	Safety	0.99	1.00	1.00	2148	
	Risk	0.98	0.94	0.96	231	0.97
Road 2 (Video 10)	Safety	0.99	1.00	0.99	876	
	Risk	0.99	0.94	0.97	139	0.97
Highway 1 (Video 8)	Safety	0.96	0.99	0.97	1509	
	Risk	0.89	0.66	0.76	198	0.82
Highway 2 (Video 20)	Safety	0.99	0.97	0.98	8915	
	Risk	0.80	0.90	0.85	1022	0.94

Figure 10
ROC and AUC for each scenario



scenario with a lower recall value, vehicle occlusion in the video may have affected distance estimation performance, leading to incorrect predictions.

In summary, the average AUC for predicting moving vehicles with the proposed model is 0.95, indicating its ability to accurately classify vehicles into safe and risky categories and issue alerts. Additionally, even in scenarios where there is a highly imbalanced distribution between risky and safe vehicles, the proposed system is still able to identify risky vehicles, demonstrating the robustness of the model.

4.5. Test dataset

To evaluate our method in a real-world environment, we took a video as the test dataset. The scenario is user facing the road with camera towards the oncoming vehicles. The camera height is set as 0.8 m. The latitude, longitude, and magnetic field value are 3.1056, 101.6764, and 329.1700 respectively. The unit pixel area is 0.64 μm and CMOS is 1/1.67. The video resolution is 1920 × 1080, the frame rate is 30 fps, and the intrinsic matrix is [[1362 0 954.9823 0] [0 1362 529.6956 0] [0,0,0,1]]. For the detection and tracking model, YOLOv8n and DeepOCSort are utilized. In the test results, our model demonstrated the capability to predict nearly all potential side collisions. Nevertheless, occasional inaccuracies in the detection frame led to sharp changes in the distance between the user and the vehicle, resulting in the predicted value being directed towards the user and triggering wrong alert. These instances require additional optimization. Figure 11 provides some test outputs.

4.6. Qualitative analysis

Tapu et al. [35], Xu et al. [36] and Madake et al. [37], have conducted detailed and in-depth review on the assistance models for the visually impaired. Among the existing models, there is no one that is similar to the proposed model. Therefore, we select several models to conduct a functional comparison and analysis with the proposed model. The result is shown in Table 7. To satisfy the requirement of portability, the models usually utilize wearable device as sensing technique. The ultrasonic sensor can only capture the minimum distance within a certain range, and the valid range is relatively short. The infrared time-of-flight (TOF) distance sensor is primarily

Table 7
Comparative analysis of navigation and obstacle detection models for the visually impaired

		Detection range		Detected object motion		Application scenario			
Approach	Sensing technique	Forward	Sideward	Stationary	Dynamic	Indoor	Outdoor	Detection method	Detection distance
Katzschmann et al. [38]	Sensor belt with infrared TOF distance sensors	✓	✓	✓	✗	✓	✓	Distance data from sensors	14 m
Caraiman et al. [39]	Stereo camera, IMU	✓	✗	✓	✓	✗	✓	3D point cloud	10 m
Tian et al. [8]	RGB-D Camera, GPS	✓	✗	✓	✗	✗	✓	Depth data, YOLOv4	20 m
Meliones et al. [10]	Stereo ultrasonic sensor	✓	✗	✓	✗	✗	✓	Distance data from sensors	5 m
Bala et al. [40]	TOF distance sensors	✓	✗	✓	✗	✓	✗	Distance data from sensors	4 m
Ours	Camera, GPS, Magnetic field sensor	✓	✓	✓	✓	✗	✓	YOLOv8, DeepSort	70 m

Figure 11
Test dataset outputs

used for measuring the distance between points along a specific direction. The depth camera based on binocular vision has high accuracy. However, considering the cost, we opt for a monocular vision camera for data acquisition. For detection range, all the models take the forward direction into account and Tian et al. [8] stress the risk of the vehicle forward. Katzschmann et al. [38] set TOF sensors at each side of the sensor belt to detect possible obstacles like branches. Our model focus on the vehicle at the side, which have not accounted for in current models. For detected object motion, all models are capable to detect stationary objects. Caraiman et al. [39] use color consistency between consecutive frames to estimate dynamic regions to get better performance of 3D reconstruction and it can save the computing power. Since our model's design for predicting location, we directly assess the object motion state by evaluating the distance changes between consecutive frames. For distance detection, our model utilizes object recognition and monocular object-specific distance estimation. Therefore, its effective range surpasses that of other wearable sensing devices. However, in terms of detection accuracy, our model is lower than TOF sensors, ultrasonic sensors, and stereo cameras.

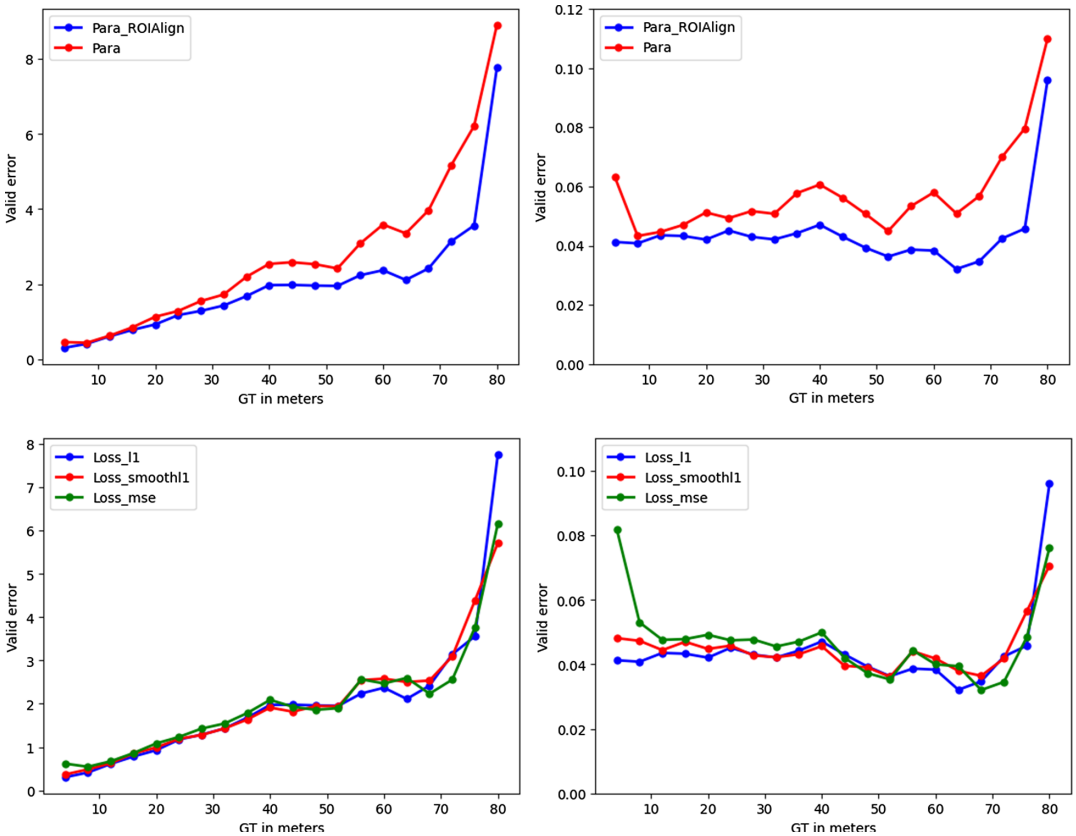
4.7. Ablation study

We compared the proposed model with different network inputs and different loss functions. In Table 8, ParaNet is trained exclusively with the bounding box parameters, while ParaROIAlignNet represents our proposed model. The employed loss function includes $L1$ loss, Smooth $L1$ loss, and MSE loss, respectively. The proposed model performs better than the ParaNet in every evaluation parameter. When comparing different loss functions, we observe that both the smooth $L1$ loss and mean squared error (MSE) loss are better than the $L1$ loss in the range of 75 to 80 meters. However, in the estimation of shorter distances, the $L1$ loss exhibits significantly better performance in terms of ALE and Abs Rel when compared to these two methods. Moreover, aside from Abs Rel, the $L1$ loss maintains optimal performance across all evaluation metrics. The ALE and Abs Rel results are shown in Figure 12.

Table 8
Comparison of proposed model trained with different input and different loss functions

	Higher is better		Lower is better		
Approach	δ<1.05	δ<1.25	ALE	Abs Rel	RMSE
ParaNet-L1	0.576	0.991	1.638	0.051	2.488
ParaROIAlignNet-L1	**0.672**	**0.994**	**1.249**	0.043	**1.924**
ParaROIAlignNet-SmoothL1	0.667	0.994	1.328	**0.043**	2.023
ParaROIAlignNet-MSE	0.626	0.994	1.390	0.047	2.030

Figure 12
The ALE and Abs Rel in meters on Kitti object dataset



5. Conclusion

In summary, this paper introduces a vision-based side collision detection system designed for visually impaired users. The system incorporates the multiple object tracking and segmentation (MOTS) model to segment and track vehicles. Additionally, the system utilizes a shallow neural network magnetic sensor for distance prediction, combining GPS, intrinsic matrices, and magnetic field sensor to calculate the real-world position of detected vehicles. Subsequently, based on the predicted real-world location sequences, the system forecasts the future locations of vehicles and performs collision predictions. The side collision detection system underwent testing on the Malaysian streets, successfully providing real-time predictions of potential collisions.

To further enhance the functionality and performance of the system, future work should address several issues. One of the future developments could be the integration of vibration-damping filtering procedures the first process in current systems. This implementation not only improves its performance and accuracy but also resists vibrations caused by user movement. In terms of location estimation, the proposed model does not dynamically filter location outliers, which reduces the accuracy of prediction to a certain extent. Improvements can be made in this regard. The last proposed avenue for future work involves integrating a traffic light recognition system. This integration would help the current system provide more accurate predictions by eliminating false alerts caused by vehicles slowing down at red lights.

Acknowledgments

We would like to express our deepest gratitude to our supervisors, Dr. Shiva and Dr. Yamani, for their invaluable guidance and support throughout the entire research process. Here are the author contributions: W.S.: Related Work, Model Design, Development, Experiment, and Evaluation; Y.S., Q.H., and J.C.: Introduction, Related Work, and Conclusion.

Conflicts of Interest

The authors declare that they have no conflicts of interest to this work.

Data Availability Statement

The data that support the findings of this study are openly available in IEEE at https://doi.org/10.1109/CVPR.2012.6248074, reference number [30].

Author Contribution Statement

Wenqing Song: Conceptualization, Methodology, Software, Data curation, Writing – original draft, Visualization, Supervision,

Project administration. **Yumeng Sun:** Validation, Investigation, Writing – review & editing. **Qixuan Huang:** Methodology, Validation, Formal analysis, Resources, Writing – review & editing. **Junyang Cheok:** Validation, Investigation, Writing – review & editing.

References

[1] Steinmetz, J. D., Bourne, R. R. A., Briant, P. S., Flaxman, S. R., Taylor, H. R., Jonas, J. B., . . . , & Vos, T. (2021). Causes of blindness and vision impairment in 2020 and trends over 30 years, and prevalence of avoidable blindness in relation to VISION 2020: The right to sight: An analysis for the global burden of disease study. *The Lancet Global Health*, *9*(2), E144–E160. https://doi.org/10.1016/S2214-109X(20)30489-7

[2] Mashiata, M., Ali, T., Das, P., Tasneem, Z., Badal, M. F. R., Sarker, S. K., . . . , & Das, S. K. (2022). Towards assisting visually impaired individuals: A review on current status and future prospects. *Biosensors and Bioelectronics: X*, *12*, 100265. https://doi.org/10.1016/j.biosx.2022.100265

[3] Giudice, N. A., Whalen, W. E., Riehle, T. H., Anderson, S. M., & Doore, S. A. (2019). Evaluation of an accessible, real-time, and infrastructure-free indoor navigation system by users who are blind in the mall of America. *Journal of Visual Impairment & Blindness*, *113*(2), 140–155. https://doi.org/10.1177/0145482X19840918

[4] Ramadhan, A. J. (2018). Wearable smart system for visually impaired people. *Sensors*, *18*(3), 843. https://doi.org/10.3390/s18030843

[5] Elmannai, W. M., & Elleithy, K. M. (2018). A highly accurate and reliable data fusion framework for guiding the visually impaired. *IEEE Access*, *6*, 33029–33054. https://doi.org/10.1109/access.2018.2817164

[6] Croce, D., Giarre, L., Pascucci, F., Tinnirello, I., Galioto, G. E., Garlisi, D., & Lo Valvo, A. (2019). An indoor and outdoor navigation system for visually impaired people. *IEEE Access*, *7*, 170406–170418. https://doi.org/10.1109/access.2019.2955046

[7] Li, Z., Song, F., Clark, B. C., Grooms, D. R., & Liu, C. (2020). A wearable device for indoor imminent danger detection and avoidance with region-based ground segmentation. *IEEE Access*, *8*, 184808–184821. https://doi.org/10.1109/access.2020.3028527

[8] Tian, S., Zheng, M., Zou, W., Li, X., & Zhang, L. (2021). Dynamic crosswalk scene understanding for the visually impaired. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, *29*, 1478–1486. https://doi.org/10.1109/TNSRE.2021.3096379

[9] Ahmed, F., Tasnim, Z., Rana, M., & Khan, M. M. (2022). Development of low cost smart cane with GPS. In *2022 IEEE World AI IoT Congress,* 715–724. https://doi.org/10.1109/AIIoT54504.2022.9817322

[10] Meliones, A., Filios, C., & Llorente, J. (2022). Reliable ultrasonic obstacle recognition for outdoor blind navigation. *Technologies*, *10*(3), 54. https://doi.org/10.3390/technologies10030054

[11] See, A. R., Sasing, B. G., & Advincula, W. D. (2022). A smartphone-based mobility assistant using depth imaging for visually impaired and blind. *Applied Sciences*, *12*(6), 2802. https://doi.org/10.3390/app12062802

[12] Bell, D., Xiao, W., & James, P. (2020). Accurate vehicle speed estimation from monocular camera footage. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *5*(2), 419–426. https://doi.org/10.5194/isprs-annals-V-2-2020-419-2020

[13] Dahl, M., & Javadi, S. (2020). Analytical modeling for a video-based vehicle speed measurement framework. *Sensors*, *20*(1), 160. https://doi.org/10.3390/s20010160

[14] Vakili, E., Shoaran, M., & Sarmadi, M. R. (2020). Single–camera vehicle speed measurement using the geometry of the imaging system. *Multimedia Tools and Applications*, *79*(27–28), 19307–19327. https://doi.org/10.1007/s11042-020-08761-5

[15] Julina, J. K. J., Sharmila, T. S., & Gladwin, S. J. (2019). Vehicle speed detection system using motion vector interpolation. In *2019 IEEE Global Conference for Advancement in Technology*, 1–5. https://doi.org/10.1109/gcat47503.2019.8978375

[16] Lee, J., Roh, S., Shin, J., & Sohn, K. (2019). Image-based learning to measure the space mean speed on a stretch of road without the need to tag images with labels. *Sensors*, *19*(5), 1227. https://doi.org/10.3390/s19051227

[17] Dong, H., Wen, M., & Yang, Z. (2019). Vehicle speed estimation based on 3D ConvNets and non-local blocks. *Future Internet*, *11*(6), 123. https://doi.org/10.3390/fi11060123

[18] Liang, H., Ma, Z., & Zhang, Q. (2022). Self-supervised object distance estimation using a monocular camera. *Sensors*, *22*(8), 2936. https://doi.org/10.3390/s22082936

[19] Lee, S., Han, K., Park, S., & Yang, X. (2022). Vehicle distance estimation from a monocular camera for advanced driver assistance systems. *Symmetry*, *14*(12), 2657. https://doi.org/10.3390/sym14122657

[20] Ding, M., Huo, Y., Yi, H., Wang, Z., Shi, J., Lu, Z., & Luo, P. (2020). Learning depth-guided convolutions for monocular 3D object detection. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11669–11678. https://doi.org/10.1109/CVPR42600.2020.01169

[21] Brazil, G., & Liu, X. (2019). M3D-RPN: Monocular 3D region proposal network for object detection. In *2019 IEEE/CVF International Conference on Computer Vision*, 9286–9295. https://doi.org/10.1109/ICCV.2019.00938

[22] Liu, Z., Wu, Z., & Tóth, R. (2020). SMOKE: Single-stage monocular 3D object detection via keypoint estimation. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 4289–4298. https://doi.org/10.1109/CVPRW50498.2020.00506

[23] Xu, H., Zhang, J., Cai, J., Rezatofighi, H., Yu, F., Tao, D., & Geiger, A. (2023). Unifying flow, stereo and depth estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *45*(11), 13941–13958. https://doi.org/10.1109/TPAMI.2023.3298645

[24] Ali, A. A., & Hussein, H. A. (2016). Distance estimation and vehicle position detection based on monocular camera. In *2016 IEEE Al-Sadeq International Conference on Multidisciplinary in IT and Communication Science and Applications*, 1–4. https://doi.org/10.1109/AIC-MITCSA.2016.7759904

[25] Bertoni, L., Kreiss, S., & Alahi, A. (2019). MonoLoco: Monocular 3D pedestrian localization and uncertainty estimation. In *2019 IEEE/CVF International Conference on Computer Vision*, 6860–6870. https://doi.org/10.1109/ICCV.2019.00696

[26] Zhu, J., & Fang, Y. (2019). Learning object-specific distance from a monocular image. In *2019 IEEE/CVF International Conference on Computer Vision*, 3838–3847. https://doi.org/10.1109/ICCV.2019.00394

[27] Zhang, Y., Ding, L., Li, Y., Lin, W., Zhao, M., Yu, X., & Zhan, Y. (2021). A regional distance regression network for

monocular object distance estimation. *Journal of Visual Communication and Image Representation*, *79*, 103224. https://doi.org/10.1016/j.jvcir.2021.103224

[28] Jocher, G., Chaurasia, A., & Qiu, J. (2023). *Ultralytics YOLO* (Version 8.0.0) [Data set]. GitHub. https://github.com/ultralytics/ultralytics

[29] Wojke, N., Bewley, A., & Paulus, D. (2017). Simple online and realtime tracking with a deep association metric. In *2017 IEEE International Conference on Image Processing,* 3645–3649. https://doi.org/10.1109/ICIP.2017.8296962

[30] Geiger, A., Lenz, P., & Urtasun, R. (2012). Are we ready for autonomous driving? The KITTI vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 3354–3361. https://doi.org/10.1109/CVPR.2012.6248074

[31] Panariello, A., Mancusi, G., Ali, F. H., Porrello, A., Calderara, S., & Cucchiara, R. (2024). Monocular per-object distance estimation with Masked Object Modeling. *Computer Vision and Image Understanding, 253*, 104303. https://doi.org/10.1016/j.cviu.2025.104303

[32] Mauri, A., Khemmar, R., Decoux, B., Haddad, M., & Boutteau, R. (2021). Real-time 3D multi-object detection and localization based on deep learning for road and railway smart mobility. *Journal of Imaging*, *7*(8), 145. https://doi.org/10.3390/jimaging7080145

[33] Jing, L., Yu, R., Kretzschmar, H., Li, K., Qi, C. R., Zhao, H., . . . , & Anguelov, D. (2022). Depth estimation matters most: Improving per-object depth estimation for monocular 3D detection and tracking. In *2022 IEEE International Conference on Robotics and Automation,* 366–373. https://doi.org/10.1109/ICRA46639.2022.9811749

[34] Vajgl, M., Hurtik, P., & Nejezchleba, T. (2022). Dist-YOLO: Fast object detection with distance estimation. *Applied Sciences*, *12*(3), 1354. https://doi.org/10.3390/app12031354

[35] Tapu, R., Mocanu, B., & Zaharia, T. (2020). Wearable assistive devices for visually impaired: A state of the art survey. *Pattern Recognition Letters*, *137*, 37–52. https://doi.org/10.1016/j.patrec.2018.10.031

[36] Xu, P., Kennedy, G. A., Zhao, F. Y., Zhang, W. J., & van Schyndel, R. (2023). Wearable obstacle avoidance electronic travel aids for blind and visually impaired individuals: A systematic review. *IEEE Access*, *11*, 66587–66613. https://doi.org/10.1109/ACCESS.2023.3285396

[37] Madake, J., Bhatlawande, S., Solanke, A., & Shilaskar, S. (2023). A qualitative and quantitative analysis of research in mobility technologies for visually impaired people. *IEEE Access*, *11*, 82496–82520. https://doi.org/10.1109/ACCESS.2023.3291074

[38] Katzschmann, R. K., Araki, B., & Rus, D. (2018). Safe local navigation for visually impaired users with a time-of-flight and haptic feedback device. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, *26*(3), 583–593. https://doi.org/10.1109/TNSRE.2018.2800665

[39] Caraiman, S., Zvoristeanu, O., Burlacu, A., & Herghelegiu, P. (2019). Stereo vision based sensory substitution for the visually impaired. *Sensors*, *19*(12), 2771. https://doi.org/10.3390/s19122771

[40] Bala, M. M., Vasundhara, D. N., Haritha, A., & Moorthy, C. V. K. N. S. N. (2023). Design, development and performance analysis of cognitive assisting aid with multi sensor fused navigation for visually impaired people. *Journal of Big Data*, *10*(1), 21. https://doi.org/10.1186/s40537-023-00689-5

How to Cite: Song, W., Sun, Y., Huang, Q., & Cheok, J. (2025). Side Collision Detection Model for Visually Impaired Using Monocular Object-Specific Distance Estimation and Multimodal Real-World Location Calculation. *Artificial Intelligence and Applications*, *3*(4), 459–472. https://doi.org/10.47852/bonviewAIA42022098