IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 11 March 2024, accepted 7 April 2024, date of publication 12 April 2024, date of current version 23 April 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3388099

RESEARCH ARTICLE

Collision Prediction in an Integrated Framework of Scenario-Based and Data-Driven Approaches

SUNGWOO LEE[1], BONGSOB SONG[1], AND JANGHO SHIN[2]
[1]Department of Mechanical Engineering, Ajou University, Suwon 16499, South Korea
[2]R&D Division, Hyundai Motor Company, Hwaseong 18280, South Korea

Corresponding author: Bongsob Song (bsong@ajou.ac.kr)

This work was supported in part by the Korea Agency for Infrastructure Technology Advancement (KAIA) grant by the Ministry of Land, Infrastructure and Transport under Grant 22AMDP-C162184-02, and in part by Hyundai Motor Company.

ABSTRACT A collision prediction framework integrating scenario-based approach with data-driven approach is proposed to enhance the safety of autonomous driving vehicles as well as advanced driver assistance systems. No matter how the autonomous driving is intelligent, it is inevitable to consider malfunction or faults of sensors, actuators, and processors, thus resulting in the collision. To address these issues, several studies have been proposed to improve performance based on model-based or data-driven approaches. However, there are several challenges in terms of the scarcity of accident data and the lack of explainability of deep neural networks. To overcome the limits of both approaches, an integrated framework that includes trajectory prediction, threat assessment, and decision-making based on convolutional neural network (CNN) for collision prediction is introduced. For more detail, both trajectory prediction based on Kalman filter and probabilistic threat metric are added in the form of a simplified bird's eye view (SBEV), which is the input to the network. In the development of the proposed algorithm, pre-crash simulation data and experimental data have been employed. A comparative study shows that the proposed algorithm outperforms the model-based algorithm on simulation data containing safety-critical scenarios. Furthermore, it outperforms the data-driven algorithm on experimental data.

INDEX TERMS Collision prediction, deep learning, risk assessment, scenario-based assessment.

I. INTRODUCTION

Active safety systems, with the potential to prevent or mitigate crashes, have contributed to an enhanced level of safety to some extent [1]. However, even highly automated vehicles are incapable of preventing accidents in situations marked by system failures or malfunctions. In May 2016, a vehicle equipped with Tesla's autonomous driving system was unable to avert a collision with a truck trailer due to a failure in object detection [2]. In March 2018, Uber's self driving vehicle could not avoid pedestrian accident due to object misclassification in Arizona [3]. In October 2023, a collision involving Cruise's robotaxi occurred, leading to the issuance of a recall for the robotaxi [4]. The National Highway Traffic Safety Administration (NHTSA) has outlined a crash scenario structure consisting of three primary components: pre-crash, impact, and injury scenarios [5]. Before the impact, autonomous vehicle (AV) endeavors to avoid collisions through the utilization of active safety systems, including automatic emergency braking (AEB) and automatic emergency steering (AES), within the crash scenario [6]. Nevertheless, in situations where collision avoidance is no longer feasible, passive safety mechanisms such as airbags and seatbelt pretensioners must be deployed to mitigate injuries. The prediction of collisions is imperative for the deployment of both active and passive safety systems.

Collision prediction studies primarily fall into two distinct categories: the model-based approach and the data-driven approach. In model-based approach, the activation of active or passive safety systems is based on threat metrics. Model-based methods can be categorized into several groups, which include single-behavior threat metrics and probabilistic approaches [7]. Single-behavior threat metrics assumes perfect measurements and relies on deterministic motion

The associate editor coordinating the review of this manuscript and approving it for publication was Mohammad S. Khan.

prediction utilizing simplified model [8], [9], [10], [11]. Some researchers have devised a collision index that examines all physically possible trajectories of the ego vehicle and surrounding objects. If the algorithm identifies a trajectory combination that would avoid a collision, then it refrains from triggering a collision detection [12]. Another single-behavior threat metrics, a predictive occupancy map (POM) was introduced to discern risks associated with multi-vehicle scenarios. Collision detection operates on the basis of the POM. When the collision risk associated with the ego vehicle, represented by the risk value at the center of the ego vehicle within the POM, surpasses a pre-determined threshold, the collision is predicted [13]. In contrast to single-behavior threat metrics, probabilistic approaches provide the advantage of considering uncertainties in state estimation during decision-making processes. The calculation of collision probability involves summing the probabilities of stochastic reachable sets or state regions corresponding to a potential collision [14], [15], [16]. Several automotive manufacturers have successfully deployed and introduced model-based algorithms to the market, specifically in the form of adaptive cruise control (ACC) and forward collision warning (FCW) systems. However, in most model-based methods, it becomes necessary to establish one or more thresholds to activate safety systems. This particular aspect can present a challenge in adapting to diverse driving scenarios, as the thresholds are frequently calibrated through heuristic methods or calculated using fixed formulas.

In the data-driven approach, multilayer perceptron neural network (MLP) was introduced for rear-end collision warning algorithm (MCWA) [17]. For the same purpose, a CNN was developed, with the input image was generated based on the gramian angular summation field (GASF) matrix. This CNN-based algorithm is denoted as the rear-end collision prediction mechanism (RCPM) [18]. Nevertheless, the majority of studies within this approach detect collisions using information derived from the primary vehicle in a lane-following scenario. Consequently, this approach may be limited, as it fails to capture the interrelationships among objects in a traffic scene [19]. To capture interrelationship, a simplified bird's eye view input representation was introduced. It can be generated from diverse sensor setups and dataset, which can enhance model's adaptability to new sensor configurations that frequently arise due to the competitive nature of the market [20]. In collision detection, data-driven approaches have demonstrated superior performance compared to model-based algorithms [17], [18], [21]. However, a data-driven approach may produce unexpected outcomes when faced with extremely rare or unknown driving scenarios that were not present in the training dataset [22]. Moreover, data-driven models necessitate safety-critical scenarios and extensive dataset for network training. However, a notable scarcity of open dataset suitable for the development of collision prediction algorithms exists.

The majority of current driving systems are designed and assessed using real-world dataset. However, occurrences of safety-critical scenarios within the dataset are infrequent. Consequently, in the realm of safety assessment, recent focus has been directed towards the exploration of scenario-based evaluation techniques and methods for generating scenarios that include safety-critical situations. In previous research, a scenario generation framework involving two main processes was introduced: scenario generation and scenario selection. During the scenario generation phase, functional, logical, or even explicitly concrete scenarios are crafted based on various sources of information such as expert knowledge, real-world driving data, and accident data. In scenario selection, scenarios are chosen by sampling from parameter ranges or distributions. For evaluation, two approaches are considered: testing-based and falsification based approaches. In the testing-based approach, safety function is evalutated based on scenarios covering parameter ranges specified by minimum and maximum values. In the falsification-based method, there are several options to discover counterexamples that violate the safety requirement, such as utilizing accident database, increasing the criticality and complexity of scenarios [23].

Several studies have explored the enhancement of the performance of passive safety systems through the utilization of pre-crash information, extending beyond mere collision prediction. Based on the identification of crash types, the activation of reversible restrains or airbags is determined accordingly. The assurance of reliable discrimination of crash types enhances the robustness and performance of passive safety systems [24], [25].

Both model-based and data-driven algorithms for collision prediction possess limitations respectively, thus highlighting the necessity for an integrated framework to enhance performance. Moreover, a review of previous studies on collision prediction indicates a lack of focus on all-around collision prediction, primarily due to the utilization of limited scenarios. In this study, we introduce a data-driven algorithm aimed at predicting all-around collisions and identifying impact sections. To mitigate unexpected outcomes from neural networks, we incorporate model-based threat metrics into an SBEV format. These metrics, which have proven effective in commercial applications, especially in collision detection, are anticipated to offer supplementary information beneficial for classifying critical situations. The proposed algorithm is developed using a range of pre-crash scenarios involving safety-critical situations, derived from accident data statistics. Additionally, real-world data is employed to enhance robustness against false alarms, given its higher complexity compared to simulated data. In previous studies related to the integrated safety systems, pre-crash information, including front or side impact area were utilized to enhance the performance of safety systems. The all-around collision prediction with a more detailed segmentation of the

impact section has the potential to design more sophisticated active or passive safety systems.

The main contributions of this study can be summarized as follows:

- In comparison to both model-based and data-driven algorithms, our collision mode prediction algorithm (CMPA), which utilizes SBEV input based on trajectory prediction and threat metric, demonstrates improved performance in predicting collisions from all directions for both simulation and experimental data.
- The proposed CMPA is designed not only to determine the occurrence of an all-around collision but also to identify a more detailed impact section when the accident occurs.

II. PROBLEM STATEMENT

The collision prediction algorithm must possess two essential properties. Firstly, it should be capable of detecting all collisions. Secondly, it should minimize the occurrence of false alarms. The challenge inherent in collision prediction lies in the frequent contradiction between these two properties. Fig. 1 presents an overview of straight crossing paths at junction scenario. In this scenarios, the ego vehicle (depicted in white) proceeds straight through an intersection and then cuts across the path of the straight-crossing target vehicle (depicted in blue) coming from a lateral direction. As depicted in Fig. 1 (left), when the target vehicle enters the intersection, it becomes challenging to definitively ascertain whether it will collide with the ego vehicle or narrowly evade contact without actual collision. From the antecedent circumstance, the inevitability of an accident becomes apparent. However, when the incident occurs, its impact is confined to a small area on the rear left side of the ego vehicle, as illustrated in Fig. 1 (right). In order to enhance safety within the context of this illustrative scenario, it is recommended that safety system designers reduce the threshold for collision prediction, even when faced with a marginal likelihood of an inevitable collision. Conversely, this adjustment may result in an elevated incidence of false alarms.

The collision mode, depicted in Fig. 2, denotes the precise point of impact on the ego vehicle during a collision. It is defined as a grid aligned with the contour of the ego vehicle, where the width and length of the ego vehicle are divided into three and five sections, respectively. Each section is designated by a two-digit number, ranging from 11 to 53 [26].

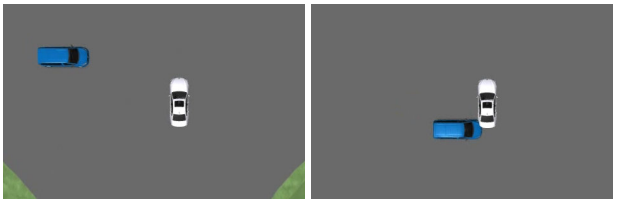

FIGURE 1. The snapshots of straight crossing paths at junction scenario.

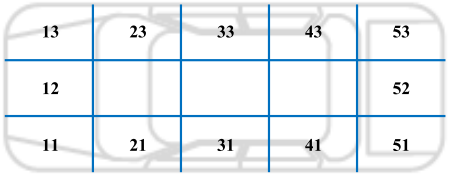

FIGURE 2. Definition of collision mode.

In this paper, we propose a CNN-based collision mode prediction algorithm that integrates a data-driven methodology with a model-based trajectory prediction and threat metric.

III. COLLISION MODE PREDICTION

The architecture of the proposed collision mode predictor is shown in Fig. 3. Sensor fusion is employed to integrate measurements from the front vision, front radar, and corner radar sensors. This integration process yields information about the states of surrounding objects with respect to the ego vehicle within the experimental data. For simulation data, information about surrounding objects is directly extracted from the simulator, with the assumption that uncertainties in the states of surrounding objects can be disregarded. In addition, data regarding the lane is conveyed to the "Abstraction" module. This information includes the lateral distance from ego vehicle to the lane markers and two degree-3 polynomials representing each lane, both derived from the front vision. The "Abstraction" module converts the current driving situation into an image suitable for input to a CNN. In Fig. 3, the DSM image depicts a scenario where the target vehicle cuts in from left on a curved road. The model-based trajectory prediction and threat metric algorithms are tasked with predicting the future states of surrounding objects and evaluating their criticality. A simplified representation of the driving scene, referred to as the dynamic semantic map (DSM), is constructed utilizing information derived from the preceding model-based algorithms and observational data. Subsequently, in the "Classification and Decision" module, a collision mode classification model utilizing CNN deduces the most probable collision mode for the current driving environment. It is crucial to ensure sensitivity and robustness in the classification process. Therefore, the determination of the collision mode is established through consideration of the likelihood of collision mode incidence, as inferred by the CNN.

A. ABSTRACTION

For early collision detection, it is essential to predict the target vehicles' behaviors. In the realm of short-term prediction within a prediction horizon of 1 to 2 s, the accuracy of trajectory prediction from model-based approaches is comparable to that achieved by data-driven methods [27]. For situations where a collision is imminent, we utilize a model-based motion predictor for short-term prediction. The trajectory prediction algorithm employs Kalman filtering with constant acceleration (CA) model to predict the future potential position of surrounding objects, along with

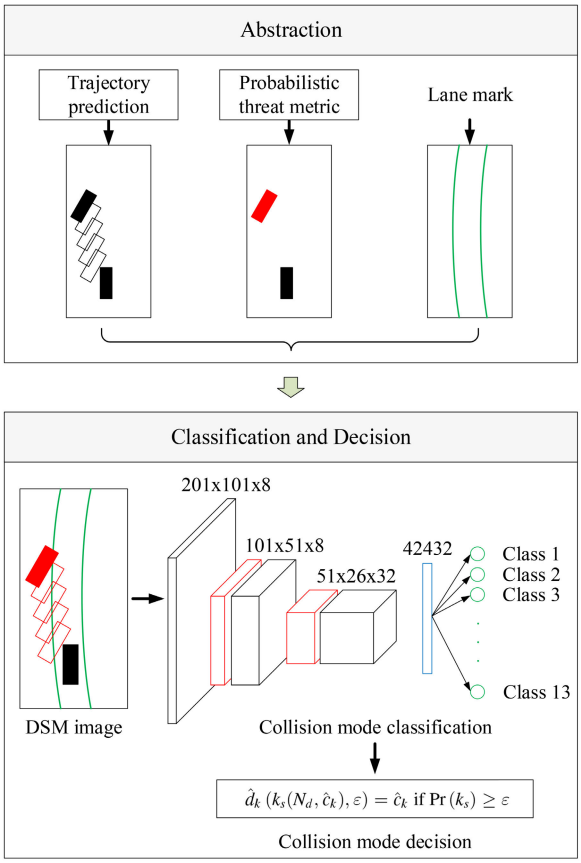

FIGURE 3. Architecture of collision mode prediction model.

estimating the error covariance [28]. In predicting the future states of objects, the accessible information encompasses the objects' current motion estimates derived from the sensor fusion algorithm. The state vector of target n are defined as follows:

$$\mathbf{x}_n = [x, y, v_x, v_y, \theta, a_x]^T \in \mathbb{R}^6 \tag{1}$$

where the subscript n corresponds to the n-th target, x denotes the relative longitudinal position, y is the relative lateral position, θ indicates the relative heading angle, v_x denotes the longitudinal velocity, v_y represents the lateral velocity, and a_x is the longitudinal acceleration.

The future states of the object can be described as a stochastic multistage process. Subsequently, the maximum likelihood prediction of the future state is computed using the Kalman filtering equations, i.e.,

$$\begin{aligned}\hat{\mathbf{x}}_n(k+1) &= \mathbf{f}(\hat{\mathbf{x}}_n(k)) + \mathbf{q}(k), \quad k = 1, \ldots, N_p \\ &\text{where} \quad \mathbf{q}(k) \sim \mathcal{N}(0, Q(k)) \end{aligned} \tag{2}$$

$$\begin{aligned}\hat{\mathbf{y}}_n(k+1) &= \mathbf{h}(\hat{\mathbf{x}}_n(k)) + \mathbf{r}(k), \quad k = 1, \ldots, N_p \\ &\text{where} \quad \mathbf{r}(k) \sim \mathcal{N}(0, R(k)) \end{aligned} \tag{3}$$

where $\mathbf{f}$ denotes the motion model, $\mathbf{q}$ represents the process noise, $\mathbf{y}$ is the measurement vector, $\mathbf{h}$ is measurement model, $\mathbf{r}$ corresponds to sensor noise, and N_p denotes the extent of the pre-established prediction time horizon, with the subscript p employed to indicate predictive. In this study, the adopted prediction time horizon spans a duration of 1 second, and the computation of prediction results occurs at intervals of 0.2 s.

To enhance the collision mode prediction model for the classification of critical situations, it would be advantageous to integrate driveability into the decision-making system. One of the most relevant metrics for assessing driveability in a given scene is the risk estimation [29]. The probabilistic approach to threat assessment has the capacity to take into account both modeling errors and measurement noise, enabling the decision making with a specific degree of confidence [7]. The calculation of collision probability involves the integration of the joint probability density function for both the ego vehicle and other objects across regions with potential collision. The collision probability can be calculated as follows:

$$g_p(t) = \max_i \left(\text{Prob}(\mathbf{x}(t+iT) \in D)\right), \quad i = 1, \cdots, N_p \tag{4}$$

where D represents the region associated with a collision, and N_p represents the duration of the pre-determined prediction time horizon.

$$\begin{aligned}&\text{Prob}(\mathbf{x}(t+iT) \in D) \\ &\quad = \iiint_{x,y,\theta \in D} p_{t+iT}(x, y, \theta | \mathbb{Y}_t)\, dxdyd\theta \end{aligned} \tag{5}$$

where $\mathbb{Y}_t$ is the cumulative set of measurement up to time t, p_{t+iT} is the probability density of the vehicles' relative position obtained from the Bayesian solution to the tracking problem [16].

The dynamic semantic map (DSM) serves as a mean to convert information including driving data, trajectory prediction, and threat metric into a SBEV. This particular representation offers the capacity to account for interactions among vehicles and exhibits flexibility in accommodating varying levels of complexity within its representation [20]. Furthermore, the SBEV representation possesses the valuable attribute of being independent from variations in dataset, such as simulation and experimental data. This quality proves advantageous when employing it to train a CNN with different data sources. DSM is created by incorporating the current state, threat metrics, and future state at time step k for all actors. In a given time frame, the DSM takes the form of an RGB image with dimensions $h \times w \times 3$. The size of the DSM has been determined to cover the $40m \times 20m$ physical region. This region is specifically situated with $30m$ extending forward from the ego vehicle and $10m$ spanning to both its left and right sides. A crucial parameter to note is the pixel resolution, which we have configured at $0.2m$. This choice has been made to strike a balance between the size of the image and its capability to accurately depict fine details. To achieve this resolution, the dimensions have been set to $h = 201$ and $w = 101$. As shown in Fig. 3, the black rectangle positioned at the bottom center of the DSM indicates the location of the ego vehicle. At any given time frame, the surrounding vehicles are rasterized

by illustrating their current position and size using filled bounding boxes. To account for the prediction of future motion of these surrounding vehicles, their bounding boxes within the prediction time horizon $[t_{k+1}, t_{k+2}, \cdots, t_{k+N_p}]$ are also rasterized. In the case of these predicted state, bounding boxes are rasterized without being filled, indicating that they are outlined shapes rather than solid-filled ones. The color assigned to surrounding vehicles is determined by probabilistic threat metric. Specifically, we allocate the probabilistic metric(g_p) to the red channel. Therefore, the level of brightness in the resulting color indicates the objects criticality or threat level. When both the trajectory prediction and threat metric are rasterized for the target vehicle, the current position of the target vehicle is indicated by the filled red rectangle, while the predicted position is shown by the empty red rectangle in the DSM image in Fig. 3. In the subsequent step, lane information is transformed into the green channel of the DSM.

B. CLASSIFICATION AND DECISION

We establish a CNN-based network denoted as f, which includes a collection of weights represented as W. This network takes a DSM image at time k, denoted as I_k, as its input. The image I_k is a three-dimensional array with dimensions $h \times w \times 3$. The primary purpose of this network is to produce an output in the form of a probability distribution. This distribution encompasses 13 distinct classes: 12 collision modes illustrated in Fig. 2 and safe class. This notation is expressed as follows:

$$\hat{y}_k = f(I_k|W) \tag{6}$$

where $\hat{y}_k \in \mathbb{R}^{13}$, $\hat{y}_k^i \in [0, 1]$ is the i-th component of $\hat{y}_k$, and $\sum_{i=1}^{13} \hat{y}_k^i = 1$.

Utilizing the DSM represented by I_k and its corresponding 13-tuple label y_k, which is one hot encoded, we proceed to train the network with the objective of determining the optimal W^* that minimize a loss function, described as follows:

$$\begin{aligned} W^* &= \arg\min_W \frac{1}{n}\sum_{k=1}^{n} \text{loss}\left(y_k, \hat{y}_k\right) \\ &= \arg\min_W \frac{1}{n}\sum_{k=1}^{n} \text{loss}\left(y_k, f\left(I_k|W\right)\right) \end{aligned} \tag{7}$$

Given that our DSM-based inputs are compact and contain limited data, there is no need for complex neural network architectures typically designed for processing natural images. Through experimentation, we have selected a network configuration. The initial two layers carry out a sequence of operations, including a convolution using 3×3 filters, a max-pooling operation with a 2×2 window, and the application of the ReLU activation function. The subsequent layer conducts a convolution without the inclusion of a max-pooling operation. Following this, there is a fully connected layer with thirteen output neurons. To prevent overfitting, dropout is incorporated into the fully connected layer [30]. The final step involves applying the soft-max function to the output of the last layer, resulting in a 13-tuple probability distribution denoted as $\hat{y}$. This distribution represents the posterior probabilities associated with collision mode.

We utilize the standard cross-entropy loss function, which is defined as follows:

$$\text{loss}\left(y_k, \hat{y}_k\right) = \frac{1}{n}\sum_{k=1}^{n}\sum_{j=1}^{13} y_k^j \log \hat{y}_k^j \tag{8}$$

where y_k^j represents the j-th element of y_k and n denotes the size of the training set.

With the trained W^*, for a given input DSM I_k, its 13-class probability distribution inference $\hat{y}_k$ is given as the feed-forward output of the network, i.e., $\hat{y}_k = f(x_k|W^*)$. The predicted collision mode $\hat{c}_k$ corresponding to I_k is the index of $\hat{y}_k$ with the maximum probability, i.e.,

$$\hat{c}_k = i^* = \arg\max_i \ \hat{y}_k^i \tag{9}$$

where $\max_i \hat{y}_k^i$ represents a component-wise maximum of $\hat{y}_k$.

Imperfect or erroneous decisions can be attributed to factors such as noisy sensor signals, the topology of the neural network, and untrained driving scenarios [22]. Considering that the decision network is not devoid of flaws, occasional imperfect decisions regarding collision modes may occur. In such circumstances, employing a testing method from the field of fault detection and isolation (FDI) can be beneficial in addressing the previously mentioned issue. We utilize a double threshold statistical testing method, which is designed to manage the probability of false alarms effectively while maximizing the detection capability in FDI. This method utilizes two separate tests with two levels of thresholds to enable the adjustment of the trade-off between detection power and the probability of false alarms [31]. It's analogous to the inherent contradiction in collision detection, where the objective is to both predict all collisions and minimize erroneous decision. To apply the double threshold testing method for making decisions regarding collision modes, we employ the second level of test from this approach. In the second level of the test, a window is introduced, and collision mode classification outcomes derived from the CNN model are accumulated. Subsequently, the number of each collision mode classification within the window is counted. Following this, the respective probabilities associated with each collision mode are computed and compared against the pre-defined threshold, ε. The final decision rule is formulated as follows:

$$\hat{d}_k\left(k_s(N_d, \hat{c}_k), \varepsilon\right) = \hat{c}_k \text{ if } \Pr\left(k_s\right) \geq \varepsilon \tag{10}$$

where $\hat{d}_k$ indicates the determined collision mode, N_d represents the window, $\hat{c}_k$ denotes classification outputs from the CNN model, and k_s is the number of respective collision mode classifications in the window.

IV. SCENARIO-BASED TRAINING

A. PRE-CRASH SCENARIO DATABASE

In this study, a database for the development of a conflict mode judgment model is created and used for training and evaluation. The database consists of simulation data and experimental data. When training a neural network to identify collision mode, it becomes imperative to incorporate pre-crash data. In the absence of an accessible open dataset comprising authentic accident driving data derived from onboard sensors in autonomous vehicles, we have undertaken the collection of scenario-based simulation data specifically intended for the development and evaluation of algorithms for decision-making regarding collision mode [32]. A collision prediction model trained solely on simulation data may exhibit inferior performance in real-world environments. Therefore, for falsification, where we search for scenarios in which the AV fails to meet the required criteria, we incorporate experimental data collected from onboard sensors in autonomous vehicles into the development of our proposed algorithm.

Simulation data acquisition process was executed using the IPG CarMaker simulation platform. In scenario-based safety assessment, a series of test scenarios is initially defined, and accident data can serve as the basis for their selection [32]. To conduct a statistical analysis of pre-crash scenarios, we utilized the crash databases of the Traffic Accident Analysis System (TAAS) in South Korea spanning from 2012 to 2014 and data from the Initiative for the Global Harmonization of Accident Data (IGLAD) covering the years 2007 to 2018 [33], [34]. From the comprehensive collection of scenarios within the statistical dataset, our specific choices included 8 non-junction scenarios and 6 junction scenarios. These scenarios were identified and given slightly modified names based on pre-crash scenario typology of NHTSA, and the corresponding list is provided in Table 1 [35], [36].

TABLE 1. Scenario catalog for simulation.

Non-junction	Junction
Vehicle following	Straight crossing paths
Vehicle cutting in	Left turn across path from opposite
Lead vehicle cutting out	Straight across path of turning left from opposite
Lead vehicle stopped	Vehicle turning left
Encroaching opposing vehicle	Straight across path of turning right from right
Vehicles changing lanes	Vehicle turning right
U-turn	-
Backing up	-

Within the selected scenarios, we generated simulation data by employing N-wise sampling for the parameter space [23]. These parameters included both stationary and trigger conditions. The stationary condition encompassed factors such as vehicle position, velocity, and acceleration, while the trigger condition included the relative position necessary to initiate specific maneuvers, such as cut-in, cut-out, and turns, for each scenario. For each of these scenarios, we collected simulation data describing the state of the vehicles. This information included details such as position, velocity, acceleration, width, length, and heading angle, and it was recorded at intervals of 0.01 s, equivalent to a frequency of 100Hz. This generated data amounts to a total of 13,750 driving records, representing a driving distance of 3,717 kilometers and a cumulative driving time of 68 hours.

The experimental data utilized in this study were acquired from a vehicle that was equipped with a variety of sensors, as depicted in Fig. 4. The sensors installed on the vehicle were used to capture information on the motion of surrounding objects in real traffic flow. Specifically, the front vision sensor was responsible for providing data related to lane markers and nearby objects, including their respective classification details. Additionally, the front and corner radar sensors were utilized to acquire data on the relative position, heading angle, velocity, and box size of the surrounding objects in local body fixed coordinates of the data collection vehicle. Furthermore, a low-cost GPS device was employed for rough precision ego localization. All the sensor data obtained were synchronized and stored on an industrial PC. Moreover, a sensor fusion algorithm was implemented to process the aforementioned sensor data and generate tracks of traffic actors, which encompass state estimates that provide more accurate information compared to the data obtained from individual sensors. In this investigation, seven different drivers operated the AV to gather data of the surrounding vehicles on both urban roads and highways in South Korea. The experimental dataset corresponds to a driving distance of 1,787 kilometers and a cumulative driving time of 25 hours. The average duration of both simulation data is approximately 20 s. In contrast, each set of raw experimental data has a duration of 2 minutes. To maintain uniformity in data length, the experimental data was divided into 20-second snippets.

The annotation of collision mode for simulation data was automatically determined by examining the information obtained from the simulation platform. Every 0.01 second, which is the sample period we set, the collision sensor within the IPG CarMaker simulator produces data indicating the occurrence or absence of a collision event. The instance when the collision sensor initially detects a collision is defined as the moment of impact (t_i). Following this, the impact area is assessed by considering the position, width, length, and heading angle at the time of the collision. The distribution of collision mode annotation is depicted in Fig. 5, while the distribution of safe annotation is presented in Table 2.

The performance of a data-driven model is notably impacted by the quality of its training data. A series of complexity measures were proposed to quantify information within driving scenarios. These measures are associated with various factors, such as the crowdedness, class diversity, and speed diversity of surrounding objects [37]. The concept of crowdedness, denoted as E^{crowd}, is the quantity of objects within a region of interest (ROI) of ego vehicle. This metric is utilized to measure the extent of congestion within a given traffic scenario. Additionally, class diversity, symbolized as

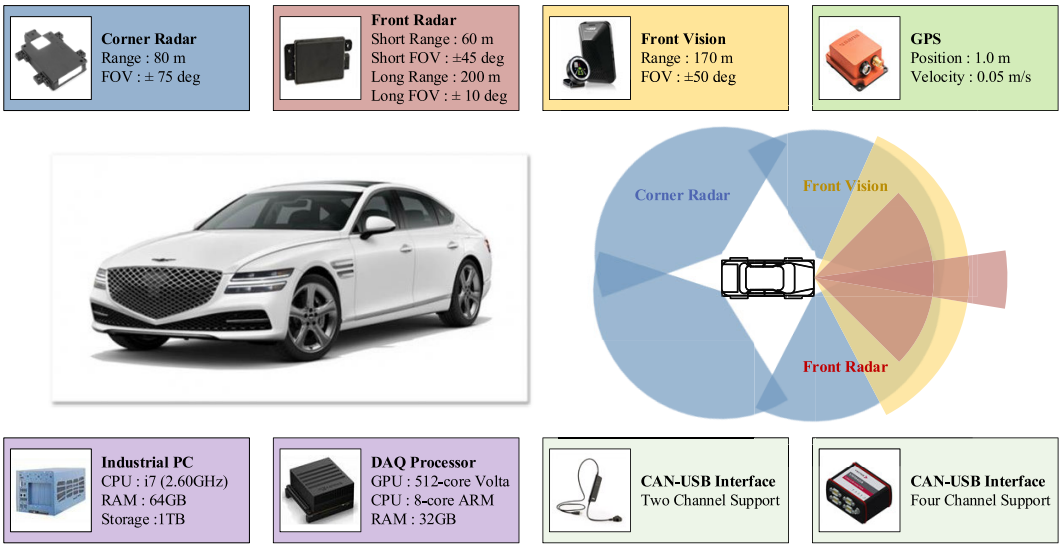

FIGURE 4. Experimental vehicle and its configuration.

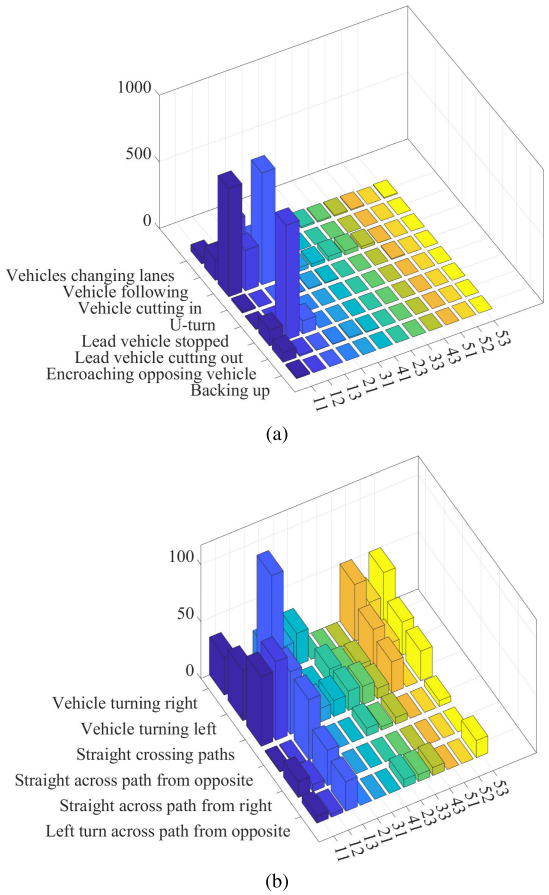

FIGURE 5. Distribution of collision mode annotation for scenario catalog, (a) Non-junction scenario, (b) Junction scenario.

E^{class}, refers to the variety of different types of actors present within a traffic scene. Speed diversity, represented by E^{speed}, denotes the variation in the speed of actors. Furthermore, we incorporate the maximum value of collision probability

TABLE 2. The number of data with safe annotation for scenario catalog.

Scenario	Number of data
Vehicles changing lanes	332
Vehicle following	750
Vehicle cutting in	1,990
U-turn	88
Lead vehicle stopped	238
Lead vehicle cutting out	796
Encroaching opposing vehicle	148
Backing up	113
Vehicle turning right	391
Vehicle turning left	961
Straight crossing paths	936
Straight across path of turning left from opposite	574
Straight across path of turning right from right	549
Left turn across path from opposite	555

concerning individual data as a threat measure, denoted as E^{threat}, to characterize the criticality of driving environment.

The distributions of complexity measures and threat measures pertaining to simulation and experimental data in pre-crash databases are depicted in Fig. 6. In the context of simulation data, the crowdedness varies from 0.8 to 1.2, class diversity ranges from 1.8 to 2, and the variation in the speed of actors spans from 0 to 380, as depicted in Fig. 6 (a) to (c). It is notable that E^{crowd}, E^{class}, and E^{speed} demonstrate distributions within a limited range when compared to experimental data. This is reasonable given that the generated simulation data typically involves one or two vehicles in a single scenario. From the perspective of threat measures, the distribution of pre-crash data in simulation leans towards high criticality, while the distribution of safe data in simulation tends towards low criticality, as illustrated in Fig. 6 (d).

Meanwhile, the three complexity measures exhibit distributions with larger values compared to those of the simulation data, as illustrated in Fig. 6 (a) to (c). Crowdedness varies from 0.8 to 6, class diversity ranges from 1.1 to 2.4, and the

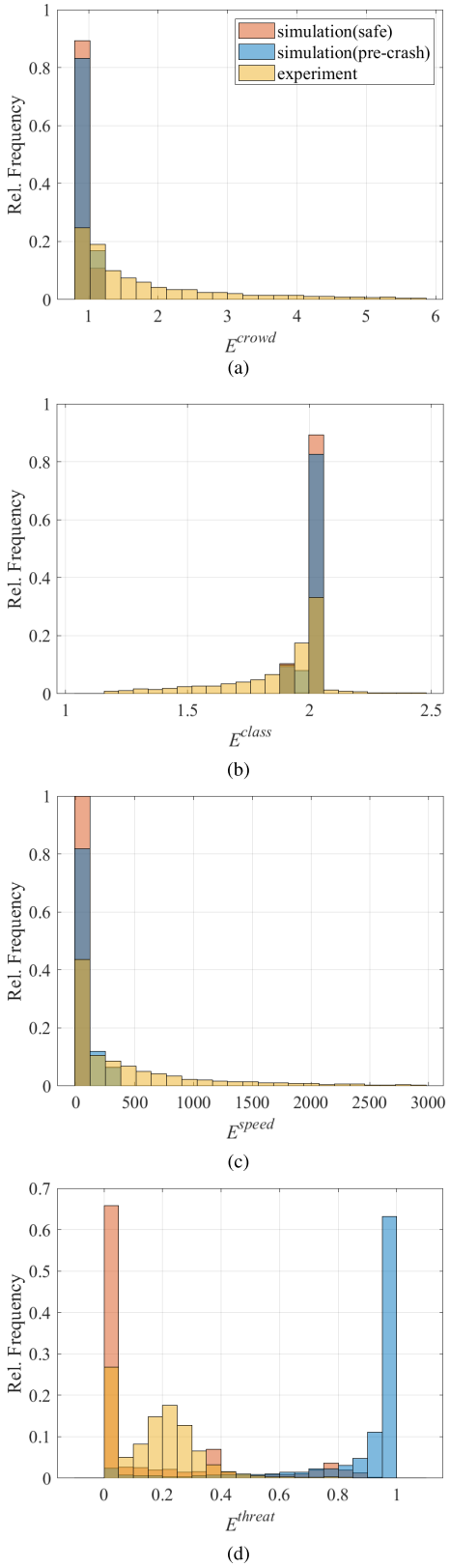

FIGURE 6. Complexity measures for pre-crash scenario database, (a) Crowdedness, (b) Class diversity, (c) Speed diversity, (d) Threat.

variation in the speed of actors spans from 0 to 3000. This discrepancy arises because experimental data is inherently more complex than simulation data, involving a greater number of objects with diverse classes and a wider variation in speed. Given that experimental data does not encompass pre-crash situations, the threat measure is distributed from 0 to 0.4, representing a lower value than the threat measure associated with simulation data in Fig. 6 (d).

B. TRAINING

We trained collision mode prediction algorithm using the active learning method. The fundamental concept behind active learning is that a machine learning algorithm can attain higher accuracy with a reduced number of training data by enabling the algorithm to actively select the data from which it learns. In the context of object detection, the active learning process typically comprises four steps: train, query, annotate, and append. In each iteration of this process, a scoring function and a sampling strategy in the query step collaboratively determine which images should be subjected to manual labeling and subsequently incorporated into the training data set [38].

In the implementation of a active learning, the data set described in previous section underwent a specific division process. Initially, the data was randomly divided into two sets: one comprising 50% for training and the other 50% for testing. Within the training data, a further subdivision took place, with 50% designated as the initial training data. The remaining 50% of the training data was reserved for subsequent iterations of the active learning process.The number of selected training and test data are summarized in Table 3.

TABLE 3. Training and test data.

	training		test	sum
	initial	remaining		
simulation data	3,177	3,253	6,356	12,786
experimental data	989	988	1,976	3,953
sum	4,166	4,241	8,332	16,739

As shown in Fig. 7, a class imbalance exists within the training data. Research literature suggests that using oversampling techniques is an effective method to address class imbalance within CNN frameworks [39]. Therefore, to address this class imbalance, oversampling technique

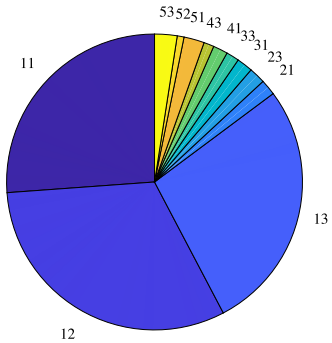

FIGURE 7. Distribution of collision mode annotation for training data.

was utilized during the initial training process, specifically targeting the minority classes linked with side and rear collisions. This involoved duplicating the number of DSM images corresponding to each collision mode within the minority class until achieving a balanced distribution.

To increase the inclusion of pre-crash data in our training set, we initially trained model utilizing only simulation data in training data. Subsequently, initially trained model was utilized in the query step to process the initial training data for both simulation and experimental data, thereby enhancing the diversity and complexity of driving scenarios. The introduction of automated annotation for collision mode decisions brought about a modification in the original image selection process, which typically relied on the scoring function in the query step, as described in [38]. Instead, the focus shifted to identifying DSM images where the network made erroneous judgments. As a result, all DSM images where the model had made incorrect judgments were included in the existing training set. The subsequent step involved training a completely new model from scratch, leveraging the appended training data set. This iterative cycle was then repeated until every piece of the training data had been considered. During each iteration of active learning, we trained the network for 38,800 iterations using the stochastic gradient descent with momentum method. We employed a batch size of 128 and initiated the learning rate at 0.01. Additionally, we reduced the learning rate by a factor of 0.1 after every 10 epochs.

C. ANALYSIS OF TRAINING SET

In Fig. 8, iteration 1 denotes the initial training data, while iteration 2 represents the appended DSM images comprising both simulation and experimental data in an active learning loop. The distribution of complexity measures and threat measures for the initial training data closely resembles the distribution of complexity and threat measures observed in the simulation data within the database, as illustrated in Fig. 8 (a) to (d).

Annotations for the appended DSM images from simulation data in iteration 2 correspond to the safe. The distribution of complexity measures for these DSM images aligns with the complexity distribution observed in the simulation data within the database. However, in Fig. 8 (d), the threat measure varies from 0.6 to 1, indicating that the added simulation data in this iteration step contains instances of high-risk data, despite being annotated as safe. The complexity measures of the appended DSM images from experimental data in iteration 2 are as follows: crowdedness ranges from 0.9 to 5.4, class diversity spans from 1.4 to 2.5, and the variation in the speed of actors spans from 0 to 3000, as depicted in Fig. 8 (a) to (c). Consequently, the inclusion of these DSM images widens the distribution of complexity measures within the training data in iteration 1.

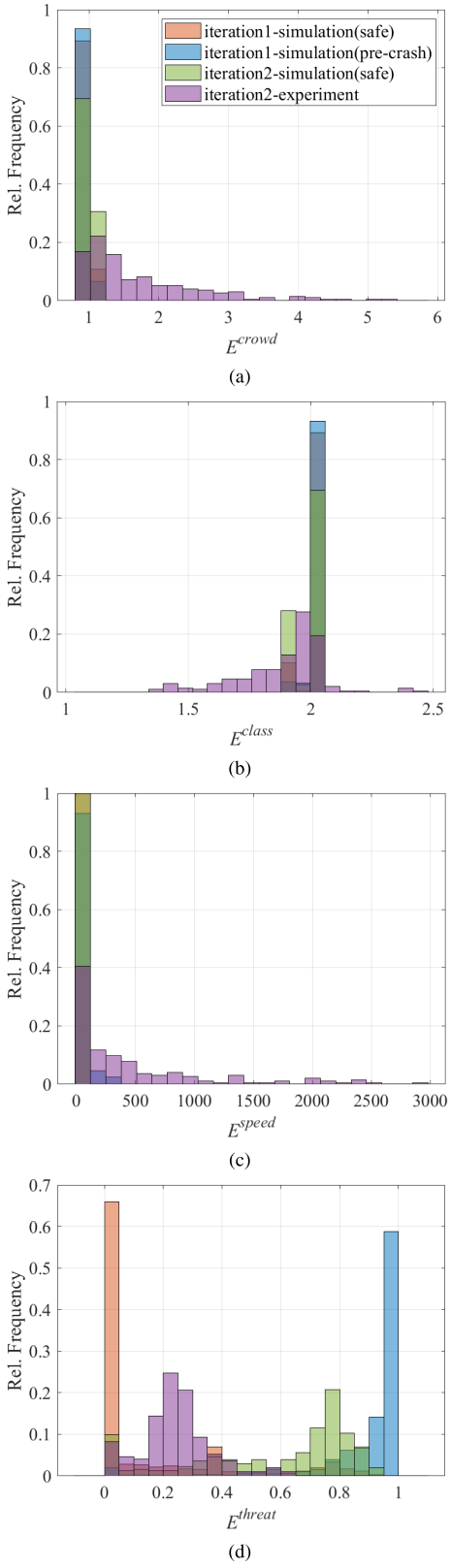

FIGURE 8. Complexity measures for training set over active learning loop, (a) crowdedness, (b) class diversity, (c) speed diversity, (d) threat.

V. TEST AND VALIDATION

We assess the performance of our proposed algorithm by comparing it with both model-based and data-driven algorithms for collision prediction. Model-based algorithms under consideration include the collision index [12],

POM [13], and collision probability [16]. Furthermore, we incorporate data-driven algorithms such as MCWA and RCPM in the comparison [17], [18]. Following this, we evaluate the effectiveness of our proposed algorithm for predicting collision modes.

To facilitate the assessment of algorithm performance for collision prediction, a confusion matrix is employed. This matrix serves to quantify the number of accurate and inaccurate predictions in comparison to the actual annotations. In this context, ''positive'' denotes a pre-crash scenario necessitating the algorithms' capacity to detect collisions, while ''negative'' represents a safe scenario characterized by the absence of a need for collision detection. Based on the comparison between the timing of impact and the detection of collisions, the results of the algorithm can be categorized into four distinct groups:

- True Positive (TP): This category encompasses situations where the algorithm successfully predicts a collision within 1.5 s prior to the moment of impact (t_i) in a pre-crash scenario.
- True Negative (TN): This category pertains to scenarios in which the algorithm correctly does not predict a collision in a safe situation.
- False Positive (FP): This group includes cases where the time difference between t_i and t_c exceeds 1.5 s ($t_i > t_c$), with t_c representing the time at which the algorithm first judges that a collision would occur.
- False Negative (FN): Within this group, we find instances where the algorithm fails to provide a collision prediction within 1.5 s before the impact, resulting in a missed prediction.

Using the four categories outlined above, the classification performance of the algorithm can be evaluated through the following metrics [40]:

- False Positive Rate (FPR): This metric represents the proportion of incorrect collision predictions out of all actual safe scenarios. It measures the algorithm's tendency to wrongly predict collisions in situations where they do not occur.

$$FPR = \frac{FP}{FP + TN} \tag{11}$$

- False Negative Rate (FNR): The False Negative Rate quantifies the fraction of incorrect predictions of safety (i.e., failure to predict a collision) out of all actual pre-crash scenarios. It assesses the algorithm's ability to correctly identify potential collisions.

$$FNR = \frac{FN}{FN + TP} \tag{12}$$

- Accuracy (ACU): Accuracy is the fraction of correct decisions made by the algorithm across all scenarios. It provides an overall measure of the algorithm's correctness in predicting both collisions and safe scenarios.

$$ACU = \frac{TP + TN}{TP + FP + TN + FN} \tag{13}$$

In addition to the performance metrics mentioned earlier, it is imperative for researchers to take into account the algorithm's capacity to predict collision at an early stage. This holds particular significance in the context of activating safety systems, where the timely anticipation of surrounding object collisions is of paramount importance. To facilitate the comparative evaluation of various collision detection algorithms, another metric comes into play. This metric is represented by the decision time (τ_c) preceding the occurrence of impact. It serves as an indicator, illustrating the algorithm's capacity to anticipate collisions in advance. Its definition is as follows:

$$\tau_c = t_i - t_c \tag{14}$$

Table 4 presents a comprehensive overview of the outcomes derived from the active learning loop. The results of iteration1 emanate from a model trained exclusively on simulation data. The accuracy in this instance attains 90.7%, accompanied by a relatively high FPR of 14.5%, in contrast to the FNR of 1.8% for the simulation data. In the case of experimental data, the accuracy reaches 86.9%, accompanied by an FPR of 13.1%. Due to the absence of unsafe situations, there are no values available for the FNR. The performance metrics for iteration2 are derived from a model trained by augmenting the initial simulation data with instances that the pre-trained model misjudged, encompassing both simulation and experimental data. Section IV-C provides insights into the initial model trained in iteration1, revealing misjudgments in simulation data where annotations indicate safety but the risk for surrounding vehicles is high, and in experimental data where the risk is relatively low but the driving situation's complexity is high. Upon appending this misjudged data to the initial training set, the accuracy demonstrates an approximately 5% increase, and the FPR exhibits an approximately 10% reduction compared to iteration1 for the simulation data. For experimental data, the accuracy experiences a 12% enhancement, accompanied by a 12% reduction in the FPR.

TABLE 4. Collision prediction performance over active learning loop.

Iter.	# DSM	Data	Accuracy (%)	False Negative Rate (%)	False Positive Rate (%)	Mean τ_c (s)
1	44,838	simulation	90.7	1.8	14.5	0.87
		experiment	86.9	-	13.1	-
2	47,315	simulation	96.3	3.6	3.7	0.87
		experiment	99.2	-	0.8	-

Fig. 9 displays the outcomes of collision prediction using the proposed algorithm and other algorithms for the scenario shown in Fig. 1. In this particular scenario, the collision event occurs at 10.8 s. In this scenario, the collision probability algorithm fails to detect a pre-crash condition before the collision event. The history of collision probability is depicted in Fig. 9 (b), with collision probability values ranging from 0 to 1. A threshold of 0.7 was applied for pre-crash decision, following the established literature. As the

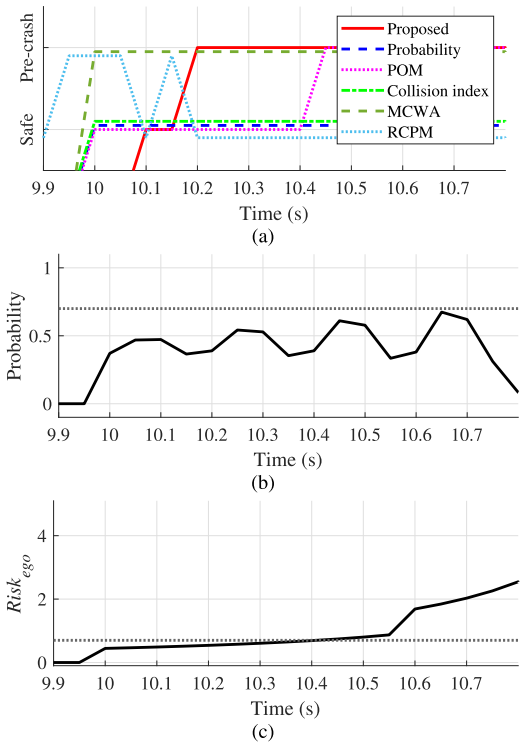

FIGURE 9. Comparison of collision prediction results for straight crossing path at junction scenario, (a) Results for the collision prediction algorithms, (b) Collision probability history, (c) The history of ego vehicle risk in POM.

ego vehicle and the target vehicle approach each other, the probability gradually increases, reaching a maximum of 0.66 at 10.65 s. However, it does not surpass the predefined threshold, resulting in a failed decision. Lowering the threshold could potentially enable the detection of collisions in this scenario, however it comes with the drawback of an increased likelihood of false alarms due to its contradictory nature.

As depicted in Fig. 9 (a), the POM successfully identified a collision event at 10.45 s. The history of ego vehicle risk, illustrated in Fig. 9 (c), ranging from 0 to 5, represents the risk value at the center of the ego vehicle in POM. For collision detection, a risk threshold of 0.7 was employed, consistent with the literature. As the two vehicles approached each other, ego vehicle risk value gradually increased and surpassed the threshold at 10.45 s.

The collision index algorithm is unsuccessful in predicting a pre-crash condition before the impact occurs. To investigate why the collision index experiences missed detections, the process of calculating the collision index at the collision moment is illustrated in Fig. 10. In this figure, the blue and red bounding boxes represent the ego vehicle and target vehicle, respectively. Two black lines extending from the front bumper of the ego vehicle represent virtual lines generated in the direction of the relative velocity vector to compute the collision index. The collision index is defined as the area of intersection between these lines and the target vehicle. However, when the impact occurs on the side of the

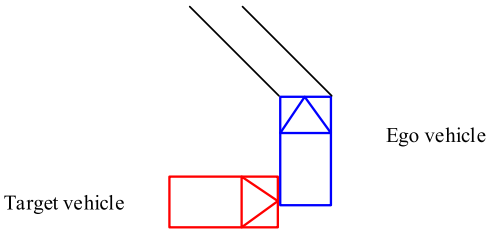

FIGURE 10. Determination of the collision index at the moment of collision for straight crossing path at junction scenario.

ego vehicle, the area of intersection cannot be calculated, as depicted in Fig. 10. Therefore, the collision index has limitations when it comes to its application in this particular pre-crash scenario.

In Fig. 9 (a), the MLP-based MCWA successfully made a pre-crash judgment at 10 s and consistently maintained pre-crash judgments until 10.8 s, which was the moment of the crash. On the other hand, the RCPM, employing a GASF matrix-based CNN model, initially predicted a pre-crash decision at 9.9 s. However, it incorrectly assessed the situation as safe from 10.2 s until the moment of the crash.

The proposed algorithm successfully makes a pre-crash assessment at 10.2 s, as demonstrated in Fig. 9 (a). DSMs at the time of the first successful assessment and at the time of the crash are presented in Fig. 11 (left). In the left DSM, the target vehicle is highlighted in red due to the application of probabilistic threat metric in the red channel. The predicted position of the target, depicted as a non-filled bounding box, contacts the side of the ego vehicle represented in black. This configuration suggests a hazardous situation with an impending collision on the left side of the ego vehicle, which the proposed algorithm identifies as a pre-crash scenario. In Fig. 11 (right), the DSM corresponding to the moment of the actual collision exhibits the target vehicle in a dark red hue. This is due to the diminished brightness of the red channel when compared to the left DSM. The reduced brightness is a consequence of the relatively small overlap between the target vehicle and the ego vehicle, resulting in a diminished probability of collision. Nevertheless, even in this circumstance, it is noteworthy that the filled bounding box representing the current position of the target vehicle

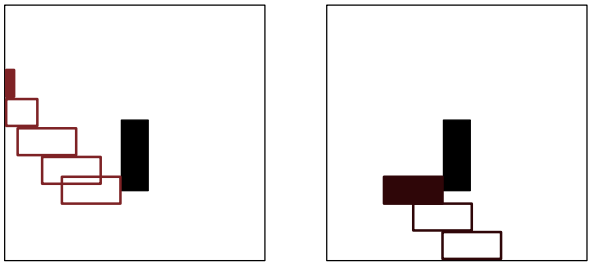

FIGURE 11. DSM images depicting instances of both successful initial decision (at 10.2s) and collision event (at 10.8s).

makes direct contact with the left side of the ego vehicle. Once again, the proposed algorithm successfully detects a pre-crash condition based on the information presented in the DSM. In this intersection scenario, the POM, MCWA, and the proposed algorithm successfully identified the collision.

TABLE 5. Comparison of collision prediction performance for simulation data.

Algorithm	Accuracy (%)	False Negative Rate (%)	False Positive Rate (%)	Mean τ_c (s)
Collision index [12]	92.7	17.4	0.2	0.26
POM [13]	71.8	21.1	33.0	0.79
Collision probability [16]	88.8	12.7	10.1	0.80
MCWA [17]	92.9	0.7	11.4	1.03
RCPM [18]	67.1	20.1	39.9	0.8
CMPA	96.3	3.6	3.7	0.87

Table 5 presents performance metrics related to collision prediction for corresponding algorithms using simulation data. As observed in Table 5, the proposed algorithm exhibits the highest accuracy, the second lowest FNR, and the second lowest FPR. Additionally, it demonstrates the earliest decision time, except for MCWA. The MCWA ranks second best among the algorithms in terms of accuracy. The collision index exhibits the third-highest accuracy, characterized by a high FNR and the lowest FPR. Across 14 pre-crash scenarios, encompassing safety-critical situations, data-driven approaches exhibit higher accuracy compared to model-based approaches, except for RCPM. This highlights the challenge faced by model-based approaches, which rely on predefined thresholds for collision detection, in adapting to diverse driving scenarios compared to data-driven approaches.

In addition to assessing simulation data, an evaluation of experimental data was conducted, as outlined in Table 6. The proposed algorithm performs similarly to the model-based algorithms and exhibits approximately 6% higher accuracy than the data-driven algorithms. For the experimental data, the model-based algorithm demonstrates superior performance compared to previous data-driven algorithms. Experimental data comprises solely safe situations, yet it is characterized by high driving situation complexity and sensor noise. This suggests that prior data-driven approaches exhibit poor robustness, as evidenced by their higher FPR when compared to model-based approaches for experimental data. On the other hand, the proposed algorithm performs

TABLE 6. Comparison of collision prediction performance for experimental data.

Algorithm	Accuracy (%)	False Negative Rate (%)	False Positive Rate (%)	Mean τ_c (s)
Collision index [12]	99.6	-	0.4	-
POM [13]	98.0	-	2.0	-
Collision probability [16]	99.2	-	0.8	-
MCWA [17]	93.3	-	6.7	-
RCPM [18]	92.9	-	7.1	-
CMPA	99.2	-	0.8	-

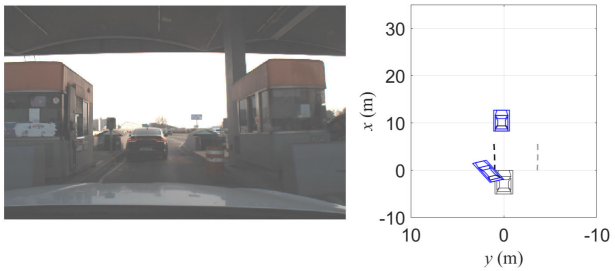

FIGURE 12. The presence of a ghost track on the tollgate structure leads to a false positive.

comparably to model-based algorithms. This indicates that it inherits the advantages of a model-based approach by utilizing a model-based methods as input. Consequently, it is more robust than other data-driven approaches, which rely solely on state variables such as position, velocity, and acceleration as input.

Given the absence of accidents during experimental data, the situation was not expected to detect collision. The majority of instances where false positives occurred can be attributed to the generation of a ghost track around structures, such as tollgates, as illustrated in Fig. 12. The occurrence of false positives is, consequently, a result of inaccurate sensor measurements. This underscores the significance of the environment perception system's ability to discern between ghost objects and real traffic actors. Consequently, the evaluation results, obtained from both simulated and experimental data, demonstrate that the proposed algorithm exhibits balanced performance in comparison to other model-based and data-driven algorithms.

In the evaluation of collision mode decisions, slightly different performance indices are employed. True positive, employed in the assessment of collision predictions, are subdivided into two distinct categories for the evaluation of collision modes: true prediction and false prediction [41].

- true prediction (tp): This category encompasses situations where the algorithm successfully identifies an impact section within 1.5 s prior to t_i in a pre-crash scenario.
- near false prediction (fp_n): In the case of a near false prediction, the algorithm foresees a collision within 1.5 s before t_i but identifies a distinct impact section adjacent to the annotation. For example, the algorithm might predict collision modes 12 when the annotated collision mode is 11.
- far false prediction(fp_f): The far false prediction category occurs when the algorithm predicts a collision but designates a different impact section more than two impact sections away from the annotation. For instance, the algorithm may predict collision mode 31 when the annotated collision mode is 11.

In accordance with the three categories specified above, as well as TN, FP, and FN, the classification performance for collision mode can be evaluated through the following metrics:

- accuracy for collision mode (acu_{cm}): The accuracy for collision mode is defined as the fraction of correct collision mode decisions made by the algorithm. This metric offers a comprehensive evaluation of the algorithm's correctness in predicting both collision modes and safe scenarios.

$$acu_{cm} = \frac{tp + TN}{tp + fp_n + fp_f + FP + TN + FN} \quad (15)$$

- relaxed accuracy for collision mode ($racu_{cm}$): The relaxed accuracy for collision mode extends the accuracy for collision mode by permitting near false predictions to be considered correct collision mode decisions within the acu metric.

$$racu_{cm} = \frac{tp + fp_n + TN}{tp + fp_n + fp_f + FP + TN + FN} \quad (16)$$

Table 7 presents the outcomes of collision mode assessment within the active learning loop. Similar to the case of collision prediction, it is evident that the results of iteration 2 surpass those of iteration 1 by 8% in terms of accuracy for collision mode and about 7% in terms of relaxed accuracy for collision mode. Table 8 illustrates the performance of class-wise accuracy for collision mode based on the results obtained from iteration 2. The relaxed class-wise accuracy for collision mode shows an improvement ranging from 6% to 18% for collision modes 11, 12, and 13, which correspond to the frontal side of the ego vehicle. In contrast, concerning the remaining collision modes, relaxed class-wise accuracy for collision mode showcases a marginally superior performance as opposed to class-wise accuracy for collision mode. The enhanced performance of relaxed accuracy in comparison to accuracy indicates that the proposed collision mode prediction algorithm forecasts the impact section as the area adjacent to the annotation, rather than misclassifying it as an area distant from the annotation when the collision mode does not precisely match the annotation.

TABLE 7. Collision mode decision performance over active learning loop.

Iteration	Number of DSM	Accuracy for collision mode (%)	Relaxed accuracy for collision mode (%)
1	44,838	76.3	87.9
2	47,315	84.3	95.4

TABLE 8. Class-wise collision mode decision performance.

Class	Accuracy for collision mode (%)	Relaxed accuracy for collision mode (%)
Collision mode 11	79.3	85.3
Collision mode 12	69.2	87.6
Collision mode 13	82.3	89.6
Collision mode 21	98.1	98.8
Collision mode 23	97.4	97.9
Collision mode 31	98.5	99.0
Collision mode 33	98.6	99.2
Collision mode 41	98.6	99.3
Collision mode 43	98.6	99.0
Collision mode 51	99.0	99.4
Collision mode 52	99.3	99.6
Collision mode 53	98.6	99.2

VI. CONCLUSION

In this study, we present a collision mode prediction algorithm that integrates CNN with a model-based threat metric and motion prediction to predict and identify potential collisions and impact section. The driving scene and model-based algorithms are reconstructed in a simplified bird's-eye view representation, which serves as the input for the CNN-based approach. The proposed algorithm's development relies on a pre-crash database, utilizing simulation data for collision mode prediction and experimental data for falsification. To train the model in this study, active learning is employed to select training data from both simulation and experimental dataset. Performance improvement is observed by augmenting the training data with instances where the model made incorrect decisions. The evaluation results indicate that the proposed algorithm demonstrates a more balanced performance compared to previously suggested model-based and data-driven algorithms for collision prediction.

In future work, other deep neural network architectures adept at processing sequential data, such as long short-term memory (LSTM), CNN-LSTM, and transformer, will be examined. Moreover, the collision mode prediction algorithm will be applied to vulnerable road user (VRU). By leveraging pre-crash information pertaining to VRU, opportunities for enhancing the performance of passive safety systems for VRU, such as pedestrian airbags and active hood lifts, can be explored.

REFERENCES

[1] J. B. Cicchino, "Effectiveness of forward collision warning and autonomous emergency braking systems in reducing front-to-rear crash rates," *Accident Anal. Prevention*, vol. 99, pp. 142–152, Feb. 2017, doi: 10.1016/j.aap.2016.11.009.

[2] National Transportation Safety Board. (2016). *Highway Accident Report, Collision Between a Car Operating with Automated Vehicle Control Systems and a Tractor-Semitrailer Truck*. [Online]. Available: https://data.ntsb.gov/Docket/?NTSBNumber=HWY16FH018

[3] National Transportation Safety Board. *Vehicle Automation Report, Tempe, Arizona*. Accessed: Nov. 5, 2019. [Online]. Available: https://data.ntsb.gov/Docket/?NTSBNumber=HWY18MH010

[4] National Highway Traffic Safety Administration. (Oct. 2023). *Part573 Safety Recall Report 23E-086*. [Online]. Available: https://static.nhtsa.gov/odi/rcl/2023/RCLRPT-23E086-7725.PDF

[5] A. M. Eigen and W. G. Najm, *Problem Definition for Pre-Crash Sensing Advanced Restraints*, document DOT HS 811 114, U.S. Department of Transportation, Apr. 2009.

[6] Euro NCAP, Leuven, Blgium. *Euro NCAP 2025 Roadmap*. Accessed: Apr. 3, 2020. [Online]. Available: https://cdn.euroncap.com/media/30700/euroncap-roadmap-2025-v4.pdf

[7] J. Dahl, G. R. de Campos, C. Olsson, and J. Fredriksson, "Collision avoidance: A literature review on threat-assessment techniques," *IEEE Trans. Intell. Vehicles*, vol. 4, no. 1, pp. 101–113, Mar. 2019.

[8] A. Doi, "Development of a rear-end collision avoidance system with automatic brake control," *JSAE Rev.*, vol. 15, no. 4, pp. 335–340, Oct. 1994.

[9] Y. Fujita, K. Akuzawa, and M. Sato, "Radar brake system," in *Proc. Annu. Meet. ITS Amer.*, vol. 1, 1995, pp. 95–101.

[10] P. Barber and N. Clarke, "Advanced collision warning systems," in *Proc. IEE Colloq. Ind. Autom. Control, Appl. Automot. Ind.*, London, U.K., 1998, pp. 2–12.

[11] E. Coelingh, A. Eidehall, and M. Bengtsson, "Collision warning with full auto brake and pedestrian detection—A practical example of automatic emergency braking," in *Proc. 13th Int. IEEE Conf. Intell. Transp. Syst.*, Sep. 2010, pp. 155–160.

[12] N. Kaempchen, B. Schiele, and K. Dietmayer, "Situation assessment of an autonomous emergency brake for arbitrary vehicle-to-vehicle collision scenarios," *IEEE Trans. Intell. Transp. Syst.*, vol. 10, no. 4, pp. 678–687, Dec. 2009.

[13] K. Lee and D. Kum, "Collision avoidance/mitigation system: Motion planning of autonomous vehicle via predictive occupancy map," *IEEE Access*, vol. 7, pp. 52846–52857, 2019.

[14] J. Jansson, J. Johansson, and F. Gustafsson, "Decision making for collision avoidance systems," presented at the SAE 2002 World Congr. Exhib., Mar. 2002, doi: 10.4271/2002-01-0403.

[15] M. Althoff, O. Stursberg, and M. Buss, "Model-based probabilistic collision detection in autonomous driving," *IEEE Trans. Intell. Transp. Syst.*, vol. 10, no. 2, pp. 299–310, Jun. 2009.

[16] J. Jansson, "Collision avoidance theory: With application to automotive collision mitigation," Ph.D. dissertation, Linköping Univ. Electron. Press, Linköping, Sweden, 2005.

[17] D. Lee and H. Yeo, "Real-time rear-end collision-warning system using a multilayer perceptron neural network," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 11, pp. 3087–3097, Nov. 2016.

[18] X. Wang, J. Liu, T. Qiu, C. Mu, C. Chen, and P. Zhou, "A real-time collision prediction mechanism with deep learning for intelligent transportation system," *IEEE Trans. Veh. Technol.*, vol. 69, no. 9, pp. 9497–9508, Sep. 2020.

[19] A. V. Malawade, S.-Y. Yu, B. Hsu, D. Muthirayan, P. P. Khargonekar, and M. A. A. Faruque, "Spatiotemporal scene-graph embedding for autonomous vehicle collision prediction," *IEEE Internet Things J.*, vol. 9, no. 12, pp. 9379–9388, Jun. 2022.

[20] S. Mozaffari, O. Y. Al-Jarrah, M. Dianati, P. A. Jennings, and A. Mouzakitis, "Deep learning-based vehicle behaviour prediction for autonomous driving applications: A review," *IEEE Trans. Intell. Transp. Syst*, vol. 23, no. 1, pp. 33–47, Aug. 2020.

[21] M. Strickland, G. Fainekos, and H. B. Amor, "Deep predictive models for collision risk assessment in autonomous driving," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 4685–4692.

[22] S. Grigorescu, B. Trasnea, T. Cocias, and G. Macesanu, "A survey of deep learning techniques for autonomous driving," *J. Field Robot.*, vol. 37, no. 3, pp. 362–386, Apr. 2020.

[23] S. Riedmaier, T. Ponn, D. Ludwig, B. Schick, and F. Diermeyer, "Survey on scenario-based safety assessment of automated vehicles," *IEEE Access*, vol. 8, pp. 87456–87477, 2020.

[24] M. Bunse, A. Kuttenberger, M. Theisen, T. Sohnke, J. S. Sangorrin, J. Hoetzel, and P. Knoll, "System architecture and algorithm for advanced passive safety by integration of surround sensing information," presented at the Soc. Automot. Eng. Int., Detroit, MI, USA, 2005.

[25] K. Cho, S. B. Choi, and H. Lee, "Design of an airbag deployment algorithm based on precrash information," *IEEE Trans. Veh. Technol.*, vol. 60, no. 4, pp. 1438–1452, May 2011.

[26] M. Wisch, "Car-to-car accidents at intersections in Europe and identification of use cases for the test and assessment of respective active vehicle safety systems," in *Proc. 26th Int. Tech. Conf. Enhanced Saf. Vehicles (ESV), Enabling Safer Tomorrow*, Eindhoven, The Netherlands, Jun. 2019, pp. 10–13.

[27] N. Deo, A. Rangesh, and M. M. Trivedi, "How would surround vehicles move? A unified framework for maneuver classification and motion prediction," *IEEE Trans. Intell. Vehicles*, vol. 3, no. 2, pp. 129–140, Jun. 2018.

[28] N. Kaempchen, K. Weiss, M. Schaefer, and K. C. J. Dietmayer, "IMM object tracking for high dynamic driving maneuvers," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2004, pp. 825–830.

[29] J. Guo, U. Kurup, and M. Shah, "Is it safe to drive? An overview of factors, metrics, and datasets for driveability assessment in autonomous driving," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 8, pp. 3135–3151, Aug. 2020.

[30] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.

[31] X. Fang, J. Gertler, M. Kunwer, J. Heron, and T. Barkana, "A double threshold-testing robust method for fault detection and isolation in dynamic systems," in *Proc. Amer. Control Conf.*, vol. 2, 1994, pp. 1979–1983.

[32] P. Junietz, W. Wachenfeld, K. Klonecki, and H. Winner, "Evaluation of different approaches to address safety validation of automated driving," in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2018, pp. 491–496.

[33] Korea Road Traffic Authority, 2016. (2016). *Traffic Accident Analysis System*. [Online]. Available: http://taas.koroad.or.kr/.

[34] (Jul. 2023). *Initiative for the Global Harmonisation of Accident Data*. [Online]. Available: http://www.iglad.net/

[35] W. G. Najm, "Pre-crash scenario typology for crash avoidance research," Nat. Highway Traffic Saf. Admin., Washington, DC, USA, Tech. Rep. DOT HS 810 767, 2007.

[36] E. Thorn, "A framework for automated driving system testable cases and scenarios," U.S. Dept. Transp., Nat. Highway Traffic Saf. Admin., Washington, DC, USA, Tech. Rep. HS 812 623, 2018.

[37] A. Sadat, S. Segal, S. Casas, J. Tu, B. Yang, R. Urtasun, and E. Yumer, "Diverse complexity measures for dataset curation in self-driving," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2021, pp. 8609–8616.

[38] E. Haussmann, M. Fenzi, K. Chitta, J. Ivanecky, H. Xu, D. Roy, A. Mittel, N. Koumchatzky, C. Farabet, and J. M. Alvarez, "Scalable active learning for object detection," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Oct. 2020, pp. 1430–1435.

[39] M. Buda, A. Maki, and M. A. Mazurowski, "A systematic study of the class imbalance problem in convolutional neural networks," *Neural Netw.*, vol. 106, pp. 249–259, Oct. 2018.

[40] R. Song and B. Li, "Surrounding vehicles' lane change maneuver prediction and detection for intelligent vehicles: A comprehensive review," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 6046–6062, Jul. 2022.

[41] A. Jain, H. S. Koppula, B. Raghavan, S. Soh, and A. Saxena, "Car that knows before you do: Anticipating maneuvers via learning temporal driving models," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 3182–3190.

SUNGWOO LEE received the B.S. and M.S. degrees in mechanical engineering from Ajou University, Suwon, South Korea, in 2016 and 2018, respectively, where he is currently pursuing the Ph.D. degree. His research interests include sensor fusion, threat assessment, and autonomous vehicles.

BONGSOB SONG received the B.S. degree in mechanical engineering from Hanyang University, Seoul, South Korea, in 1996, and the M.S. and Ph.D. degrees in mechanical engineering from the University of California at Berkeley (UC Berkeley), Berkeley, CA, USA, in 1999 and 2002, respectively. He was a Research Engineer with California Partners for Advanced Transit and Highways Program, UC Berkeley, until 2003. He is currently a Professor with the Department of AI Mobility and Mechanical Engineering, Ajou University, Suwon, South Korea. His research interests include sensor fusion, convex optimization, collision avoidance, and threat assessment with applications to intelligent vehicles.

JANGHO SHIN received the B.S. and M.S. degrees in automotive engineering from Hanyang University, Seoul, South Korea, in 1995 and 1997, respectively, and the Ph.D. degree in mechanical engineering from the University of Michigan, Ann Arbor, MI, USA, in 2007. He is currently a Senior Research Engineer with Hyundai Motor Company, Hwaseong, South Korea. His research interests include integrated safety systems, autonomous vehicle, and safety performance development.