JJEE

Volume 11 | Number 3 | September 2025 | Pages 602-618

Jordan Journal of Electrical Engineering

ISSN (print): 2409-9600, ISSN (online): 2409-9619

Homepage: jjee.ttu.edu.jo

Hybrid Deep Learning Framework with Meta-Learning for Real-Time Collision Prediction in Autonomous Systems

Uma Mahesh Babu B[1*], Giri Babu K[2], T. Krishna B[3]

[1, 3] ECE department, JNTUK Kakinada; Andhra Pradesh, India

E-mail: umamahesh.balla@staff.vce.ac.in

[2] Electronics and Communication Department in VVIT, Guntur, Andhra Pradesh, India

Received: May 09, 2025 *Revised: Jul 09, 2025* *Accepted: Sep 02, 2025* *Available online: Nov 27, 2025*

***Abstract*—** Real-time collision prediction is critical for the safety and reliability of autonomous driving systems. However, accurately forecasting collisions remains challenging due to the complexity of dynamic driving environments, noisy sensor data, scarcity of rare collision events, limited adaptability to new scenarios, and strict real-time constraints. Existing collision prediction models typically process spatial and temporal data separately, struggle to filter sensor noise effectively, and require extensive retraining for new conditions, hindering practical real-time deployment. This paper introduces Hybrid Deep Collision Prediction Network (HDC-Net), a novel unified framework designed to overcome these challenges. HDC-Net integrates a dilated convolutional neural network (CNN) with a dual-branch recurrent neural network (RNN) comprising Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) modules to jointly capture spatial context and multi-scale temporal dynamics. A self-attention mechanism is embedded to filter sensor noise and highlight essential collision indicators. Additionally, HDC-Net employs Model-Agnostic Meta-Learning (MAML) to facilitate rapid model adaptation to new driving conditions and leverages generative adversarial networks (GAN) for synthesizing realistic collision scenarios, addressing data scarcity. Computational optimizations, including hierarchical attention pooling and kernel fusion, ensure the model's real-time operability. The performance of HDC-Net was evaluated on the DeepAccident dataset using a rigorous 4-fold cross-validation. Results demonstrate that HDC-Net achieves a collision prediction accuracy of 89.3%, a time-to-collision error of 0.42 seconds, a trajectory deviation of 0.17 meters, and an inference speed of 18.4 ms per frame. Compared to state-of-the-art baselines, HDC-Net significantly improves prediction accuracy by approximately 4.6% while maintaining efficient real-time performance.

***Keywords*—** Collision detection; Trajectory tracking; Deep learning; Meta-learning; Motion prediction; Autonomous vehicles; Sensor fusion; Computational optimization.

1. INTRODUCTION

Real-time collision prediction is critical for autonomous driving safety and advanced driver-assistance systems, as early warnings enable evasive actions to prevent crashes. However, accurately anticipating collisions in complex and dynamic road environments—populated with numerous moving vehicles, cyclists, and pedestrians under constantly changing conditions—remains extremely challenging. Existing collision prediction approaches have significant limitations. Specifically, many models process spatial and temporal information separately (e.g., using CNN-based vision modules and distinct RNN-based temporal modules), which can miss important spatiotemporal interactions [1, 2]. These models are also often vulnerable to sensor noise and environmental variation (changes in lighting, weather, or sensor calibration can lead to unstable or false predictions) [3]. Moreover, training data for severe crashes is extremely limited because such dangerous events are rare

DOI: https://doi.org/0.5455/jjee.204-1746791957

in real-world driving, hindering models from learning robust collision patterns [4]. Models typically struggle to adapt to new scenarios or unseen road conditions without extensive retraining, and complex architectures can be too slow for real-time deployment on vehicles. Therefore, a new approach is required to overcome these challenges. The ideal collision prediction model should jointly capture spatial and temporal features, filter out irrelevant noise, learn effectively from limited examples, and run within real-time constraints. However, few – if any – existing methods satisfy all of these requirements simultaneously, which defines a clear research gap that this work aims to fill.

To address this gap, a unified framework called Hybrid Deep Collision Prediction Network (HDC-Net) is proposed. HDC-Net integrates multiple advanced components: it fuses spatial and temporal processing via a hybrid architecture that combines a dilated CNN with a dual-branch recurrent encoder (utilizing long short-term memory (LSTM) and gated recurrent unit (GRU) modules in parallel) to capture multi-scale motion dynamics, and it employs a self-attention mechanism to emphasize salient features while suppressing noise. The model further incorporates a meta-learning algorithm, Model-Agnostic Meta-Learning (MAML) [5], for rapid few-shot adaptation to new driving environments. In addition, HDC-Net leverages a generative adversarial network (GAN) [6] to augment the scarce collision data with realistic synthetic examples, and it includes architectural optimizations (such as hierarchical attention pooling and convolutional kernel fusion) to ensure efficient real-time inference.

The key contributions of this work are summarized as follows:

- Dual-Branch Spatiotemporal Encoder: A novel two-stream recurrent module (LSTM for long-term patterns and GRU for short-term changes) that simultaneously captures multi-scale motion dynamics from input sequences, improving the detection of both gradual and sudden impending collisions.
- Noise-Filtering Attention Mechanism: Integration of a self-attention layer that highlights critical features (e.g. imminent threat cues) and filters out sensor noise or irrelevant background, leading to more robust collision predictions under varying environmental conditions.
- Meta-Learning for Adaptation: Application of a MAML-based meta-learning strategy that enables the model to quickly adapt to new scenarios or rare crash situations with only a few training samples, significantly enhancing generalization to unseen driving conditions.
- Synthetic Data Augmentation: Use of a GAN-generated synthetic collision dataset to supplement real training data, which alleviates the data scarcity problem by providing diverse realistic crash examples and thus improves the model's ability to recognize rare collision events.
- Real-Time Optimization: Deployment of efficient inference optimizations, including hierarchical attention pooling and convolutional kernel fusion, which reduce computational load and latency. As a result, HDC-Net can operate in real time (processing a frame in approximately 18 ms) on typical autonomous vehicle hardware, satisfying on-board timing requirements.

Finally, the remainder of the paper is organized as follows. Section 2 reviews related work on collision prediction and situates HDC-Net in the context of existing spatiotemporal modeling approaches, meta-learning techniques, and real-time systems. Section 3 details the HDC-Net architecture and its components, including the dual-branch encoder, attention mechanism, meta-learning integration, and optimization techniques. Section 4 describes the experimental study, covering the dataset, evaluation metrics, and baseline models used for comparison. Section 4 presents the results and analysis, comparing HDC-Net to other approaches and reporting ablation studies to examine the contribution of each component. Section 4 offers a discussion on the implications of the results, the model's behavior in various scenarios, and current limitations. Section 5 concludes the paper and outlines potential directions for future research.

2. RELATED WORK

Research in collision prediction covers several essential areas, including spatiotemporal modeling, attention-based noise reduction, few-shot and meta-learning techniques, and computational efficiency for real-time inference. Each area provides valuable insights but also reveals critical limitations that highlight the need for an integrated approach.

Several studies have focused on combining spatial and temporal features to predict vehicle trajectories and potential collisions. Li, Haichuan et al. [7] proposed the BCSSN model, which employs a bi-directional compact spatial-separable network to integrate CNN-derived spatial data and temporal features via LSTM. This approach demonstrated improved predictive performance over purely spatial models but still lacked mechanisms for rapid adaptation to new driving scenarios. Similarly, Geethanjali and Valarmathi introduced IChOA-CNN-LSTM [8], which combined CNN-extracted features with LSTM sequences and optimized these interactions using chaotic optimization algorithms. Despite their advancements in modality fusion, these models generally require substantial retraining when introduced to new conditions, limiting their practicality for adaptive driving scenarios. Gao et al. [9] further demonstrated that hybrid deep learning models integrating CNN and LSTM effectively predict vehicle behavior over extended periods. Nevertheless, these models typically assume consistent environmental conditions and data availability, reducing their generalizability and adaptability.

Sensor noise and irrelevant environmental details significantly affect collision prediction accuracy. To address these issues, attention mechanisms have been integrated into prediction models. Xu et al. [10] proposed a trajectory prediction method based on CNN, Bi-directional LSTM, and multi-head attention, significantly enhancing the model's focus on critical information. Similarly, Zhao et al. [11] introduced a CNN-LSTM-Attention model tailored specifically for identifying near-crash events, which demonstrated robustness against noisy mountain road conditions. Additionally, Malawade et al. [12] leveraged scene-graph embeddings combined with spatiotemporal attention, effectively capturing complex interactions among various road objects to enhance collision prediction accuracy. While these attention-based models effectively isolate essential predictive cues, they do not incorporate meta-learning approaches, limiting their flexibility when exposed to novel scenarios or significant distribution shifts in driving conditions.

Rapid adaptation to unseen driving environments with limited new data is a crucial capability for autonomous systems. Meta-learning techniques, such as Model-Agnostic Meta-Learning (MAML) introduced by Pawar et al. [13], have shown promise for fast adaptation across tasks with minimal retraining. Zhang et al. [14] demonstrated the effectiveness of meta-learning with their Meta-ZSDETR model, achieving zero-shot object detection for autonomous driving scenarios. Similarly, Li et al. [15] proposed a hybrid deep-learning framework incorporating meta-learning to enhance object detection and recognition accuracy. Despite the success of these meta-learning approaches in related computer vision tasks, their direct application to collision prediction remains largely unexplored. Current meta-learning research has yet to thoroughly address rapid model adaptation specifically tailored for real-time collision prediction in autonomous driving, leaving a critical gap that necessitates further exploration.

Real-time performance is vital for practical deployment in autonomous driving scenarios, prompting research efforts focused explicitly on computational efficiency. Liu et al. [16] presented a real-time regional collision-risk prediction model based on recurrent neural networks, explicitly optimized for low inference latency in maritime environments. Seo and Jung [17] similarly developed a robust collision prediction system optimized for autonomous delivery robots, achieving rapid processing times while maintaining adequate prediction accuracy. However, many efficient models often compromise prediction reliability or adaptability to reduce computational complexity. Thus, there remains a notable challenge in balancing high predictive accuracy, adaptability to new conditions, and strict real-time processing constraints simultaneously in one cohesive model.

Collectively, prior studies offer significant advancements in individual aspects of collision prediction, such as effective spatiotemporal integration [9, 8], attention-driven noise reduction [10-12], meta-learning for adaptability [13-15], and computational optimization [17, 16]. However, existing frameworks typically address these challenges independently rather than integrating them cohesively. Models that successfully fuse spatial-temporal features frequently lack adaptability to new scenarios. Attention-based methods effectively filter noise but do not incorporate rapid adaptation mechanisms. Meta-learning models enhance adaptability but have not yet specifically addressed real-time collision prediction. Efficiency-oriented approaches often sacrifice predictive performance or adaptability for computational speed.

HDC-Net seeks to overcome these limitations by unifying spatiotemporal modeling, noise-aware attention mechanisms, meta-learning for rapid adaptation, and computational optimizations into a single, comprehensive framework. This holistic integration represents a significant advancement beyond existing methods, specifically designed to address the multifaceted requirements of real-time collision prediction in dynamic and unpredictable autonomous driving scenarios.

3. METHODS AND MATERIALS

This section describes the architecture and workflow of HDC-Net for real-time collision prediction in autonomous driving. Fig. 1 illustrates the overall framework of the proposed Hybrid Deep Collision Prediction Network (HDC-Net). The model integrates spatial feature extraction and temporal sequence modeling using a carefully structured approach.

Initially, sensor data from cameras, LiDAR, and radar are synchronized and preprocessed. Spatial features are extracted using a Dilated CNN-based Spatial-Semantic Encoder, which captures environmental context effectively through dilated convolutions. Simultaneously, a Dual-Branch Temporal Encoder processes temporal features. This encoder uses two parallel recurrent neural networks: Long Short-Term Memory (LSTM) to model long-term temporal dependencies, and Gated Recurrent Unit (GRU) to capture short-term dynamic variations.

The spatial and temporal features extracted by these modules are subsequently combined in a self-attention fusion module. This module selectively highlights critical collision-related indicators while reducing the influence of noisy sensor inputs. Hierarchical attention pooling and kernel fusion further compress and optimize the combined embeddings, improving computational efficiency for real-time inference. Predictions including collision probability, time-to-collision, and trajectory deviations are then generated by a dedicated Scatter-Sum Mixture Head, as clearly shown in Fig. 1.

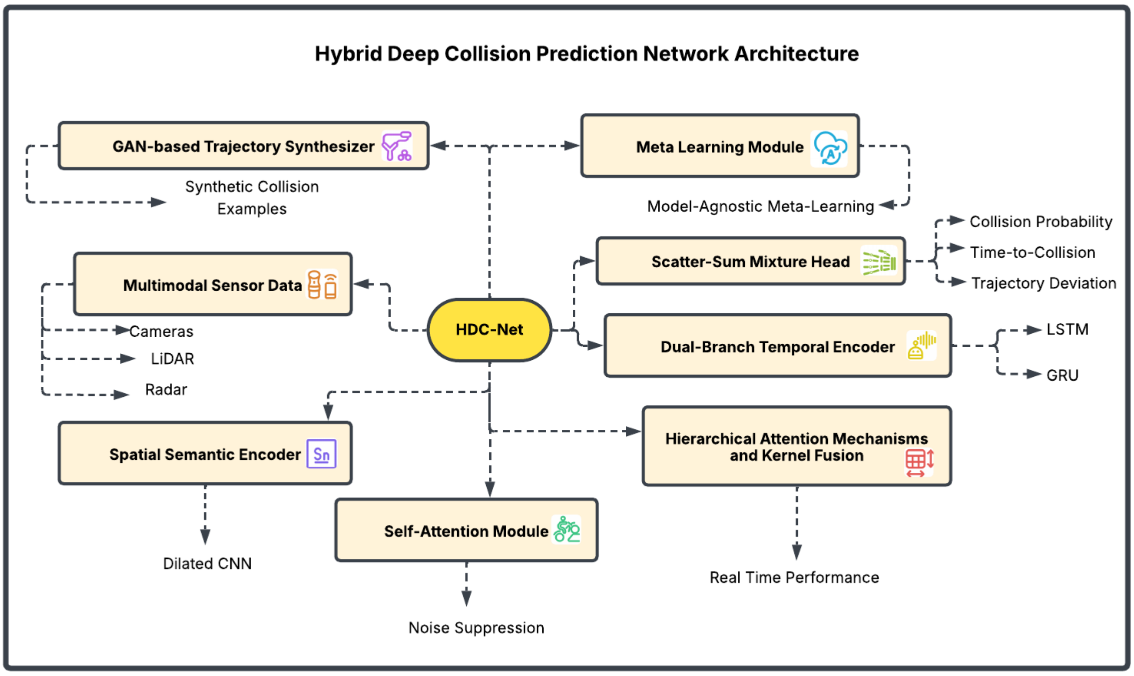

Fig. 1. The architecture diagram of the HDC-net framework.

During the training phase, additional methods support effective model learning. A GAN-based Trajectory Synthesizer augments limited real-world crash data by generating realistic synthetic collision scenarios. Moreover, the integration of Model-Agnostic Meta-Learning (MAML) enables rapid adaptation of the model parameters to new driving situations using minimal data. Fig. 1 depicts these training-specific components as dashed lines, clearly distinguishing training from inference processes. Overall, this structured architecture ensures accurate collision prediction while maintaining real-time performance and adaptability to changing driving scenarios.

3.1. Dual-Branch Sequence Encoder

The Dual-Branch Sequence Encoder simultaneously employs two parallel recurrent neural network modules—Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU)—to effectively capture motion dynamics at different temporal scales. The LSTM branch specifically addresses long-term temporal dependencies, ensuring that extended motion patterns are accurately tracked. In parallel, the GRU branch captures short-term dynamics, enabling responsiveness to rapid, transient changes in object trajectories.

At each discrete timestep t, both the LSTM and GRU receive an identical input feature vector x_t. The LSTM processes these inputs through its internal gating mechanisms, updating its hidden state h_t and memory cell c_t via the following equations:

Forget Gate: Eq. 1

$$f_t = \sigma\left(W_f x_t + U_f h_{t-1} + b_f\right) \quad (1)$$

Input Gate: Eq. 2

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \quad (2)$$

Candidate Cell State: Eq. 3

$$\tilde{c}_t = tanh(W_c x_t + U_c h_{t-1} + b_c) \quad (3)$$

Updated Cell State: Eq. 4

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \quad (3)$$

Output Gate: Eq. 5

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \quad (4)$$

Hidden State Output: Eq. 6

$$h_t = o_t \odot tanh(c_t) \quad (5)$$

Here, the variables and parameters are defined as follows: x_t: Input vector at time t , h_t: Hidden state at time t , c_t: Cell state at time t , f_t, i_t, o_t: Forget, input, and output gates, respectively , W_f, W_i, W_c, W_o: Weight matrices for input vectors , U_f, U_i, U_c, U_o: Weight matrices for hidden states , b_f, b_i, b_c, b_o: Bias vectors , σ: Sigmoid activation function , tanh: Hyperbolic tangent activation function , $\odot$: Element-wise multiplication operator.

The GRU simultaneously processes the input x_t and calculates its hidden state h'_t using the following gating mechanisms:

Update Gate: Eq. 7

$$z_t = \sigma(W_z x_t + U_z h'_{t-1} + b_z) \quad (6)$$

Reset Gate: Eq. 8

$$r_t = \sigma(W_r x_t + U_r h'_{t-1} + b_r) \quad (7)$$

Candidate Hidden State: Eq. 9

$$\tilde{h}'_t = tanh(W_h x_t + U_h(r_t \odot h'_{t-1}) + b_h) \quad (8)$$

Hidden State Update: Eq. 10

$$h'_t = (1 - z_t) \odot h'_{t-1} + z_t \odot \tilde{h}'_t \quad (10)$$

Here, the symbols represent: h'_t: GRU hidden state at time t , z_t, r_t: Update and reset gates, respectively, W_z, W_r, W_h: Input weight matrices , U_z, U_r, U_h: Hidden state weight matrices , b_z, b_r, b_h: Bias vectors. The outputs from the LSTM and GRU branches, h_t and h'_t, are then integrated to produce the unified hidden representation h_t^{dual}.

Specifically, this integration is performed by concatenating the two hidden states into a single vector, as formally represented by: Eq. 11

$$h_t^{dual} = \left[h_t\ ;\ h_t^{'}\right] \quad (9)$$

In this notation, [· ; ·]: Concatenation operator, merging both hidden states into one extended feature representation. By combining both branches through concatenation, the Dual-Branch Sequence Encoder ensures that comprehensive temporal information—spanning immediate transitions and prolonged dynamics—is simultaneously captured, thus enhancing the predictive capability of the overall model architecture.

3.2. Attention Mechanisms

The self-attention mechanism in HDC-Net selectively emphasizes features crucial for collision prediction, effectively filtering irrelevant sensor noise and highlighting critical indicators of potential collisions. This mechanism ensures robust integration of spatial and temporal embeddings obtained from preceding encoders, as illustrated explicitly in Fig. 1.

Specifically, HDC-Net employs scaled dot-product self-attention. In this process, attention scores are computed from input embeddings transformed into three distinct representations: queries (Q), keys (K), and values (V). The queries represent embeddings seeking contextual relevance; keys correspond to embeddings that determine the relevance to these queries; and values represent embeddings providing the actual information content. The scaled dot-product attention is mathematically formulated as follows:

Step 1 (Compute Attention Scores): Eq. 12

$$Attention(Q,K,V) = softmax\left(\frac{QK^{\top}}{\sqrt{d_k}}\right)V \tag{10}$$

where, $Q \in \mathbb{R}^{n\times d_k}$: Query matrix derived from the embedding inputs, where n is the sequence length and d_k is the dimensionality of the queries, $K \in \mathbb{R}^{n\times d_k}$: Key matrix derived similarly from embedding inputs, $V \in \mathbb{R}^{n\times d_v}$: Value matrix containing the information to be selectively weighted and aggregated, with dimensionality d_v. , d_k: Dimensionality of the queries and keys, serving as the scaling factor to maintain stable gradient magnitudes.

Step 2 (Definition of Symbols): Q, K, V : Queries, Keys, and Values matrices respectively, d_k: Dimension of queries and keys, acting as scaling factor, n: Sequence length, denoting the number of embedding vectors processed simultaneously, softmax(·): Softmax activation function, applied row-wise to normalize attention scores.

Within the HDC-Net architecture (as shown explicitly in Fig. 1), this self-attention module fuses outputs from the Spatial-Semantic Encoder (CNN-based spatial features) and the Dual-Branch Sequence Encoder (LSTM and GRU temporal features). Through this integration, the attention mechanism generates refined embeddings that prioritize contextually important features and suppress sensor-induced noise and irrelevant temporal variations, thus significantly enhancing the reliability and accuracy of collision prediction outcomes.

3.3. Spatial-Semantic Encoding

Dilated convolutions are specifically utilized within the Spatial-Semantic Encoder module of HDC-Net to achieve a broader receptive field without sacrificing spatial resolution. This broader receptive capability is essential for accurately capturing contextual spatial relationships and identifying subtle collision indicators within complex driving environments.

Formally, the dilated convolution operation implemented in HDC-Net is defined by the following equation: Eq. 13

$$Y(i,j) = \sum_m \sum_n X(i + d \cdot m, j + d \cdot n) \cdot K(m,n) \tag{11}$$

In this equation, the variables are explicitly defined as follows:

X : Input feature map tensor, Y : Output feature map tensor after convolution, K: Convolution kernel with size (m,n),d: Dilation rate (integer factor), controlling the spacing between kernel elements, thereby enlarging the receptive field without reducing spatial resolution, (i,j): Spatial coordinates within the output feature map, (m,n): Indices iterating over the kernel dimensions.

Following convolution, a Rectified Linear Unit (ReLU) activation function is applied elementwise to introduce non-linearity into the feature extraction process, as shown in Eq. 14.

$$F = ReLU(Y) = max(0,Y) \tag{12}$$

Here, F: Activated output feature map tensor, ReLU$(\cdot)$: Rectified Linear Unit activation function.

The spatial-semantic features extracted by this dilated convolutional encoding module (F) are subsequently integrated with temporal features generated by the Dual-Branch Sequence Encoder via the self-attention mechanism, as illustrated explicitly in Fig. 1. This integration step produces refined, noise-filtered embeddings that feed into the subsequent Scatter-Sum Mixture Head module, thus directly contributing to accurate real-time collision prediction outcomes.

3.4. Meta-Learning Integration

Model-Agnostic Meta-Learning (MAML) is integrated into HDC-Net to enable rapid adaptation of model parameters to new, unseen collision scenarios with minimal retraining. This capability is particularly critical given the unpredictable and diverse driving conditions encountered in real-world autonomous vehicle operations.

Within the context of HDC-Net, tasks for MAML are explicitly defined as distinct subsets of collision scenarios or environmental conditions extracted from the training dataset. For each task $\mathcal{T}_i$, data is partitioned into two distinct subsets:

Support Set ($\mathcal{D}_i^{\text{support}}$): A limited subset of labeled examples used for performing task specific model parameter adaptation.

Query Set ($\mathcal{D}_i^{\text{query}}$): Another subset of labeled examples utilized to evaluate and optimize the generalization capability of adapted parameters.

MAML training occurs through a two-step process comprising inner-loop (task-specific adaptation) and outer-loop (global parameter update):

Inner-Loop Adaptation (Task-Specific Update): Given initial model parameters θ, a task-specific adaptation for task $\mathcal{T}_i$ is computed via gradient descent on the support set loss function: Eq. 15.

$$\theta'_i = \theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_i}\left(\theta, \mathcal{D}_i^{support}\right) \tag{13}$$

where, θ: Global initial parameters of the HDC-Net model, θ'_i: Adapted parameters specific to task $\mathcal{T}_i$, α: Inner-loop learning rate (task-specific adaptation rate), $\mathcal{L}_{\mathcal{T}_i}$: Loss function for task $\mathcal{T}_i$, typically defined as cross-entropy or mean squared error, depending on the prediction objective.

Outer-Loop Optimization (Global Parameter Update): The global model parameters θ are optimized based on the adapted parameters across multiple tasks by aggregating gradients computed using the query sets: Eq. 16.

$$\theta \leftarrow \theta - \beta \nabla_\theta \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \quad \mathcal{L}_{\mathcal{T}_i}\left(\theta'_i, \mathcal{D}_i^{query}\right) \tag{14}$$

where, β: Outer-loop learning rate (global update rate), $p(\mathcal{T})$: Distribution of tasks sampled during meta-training.

Practically, this meta-learning approach ensures that HDC-Net can quickly adapt its predictive parameters when encountering previously unseen collision scenarios or novel driving conditions. For instance, upon encountering new collision types or unfamiliar driving environments, the model rapidly fine-tunes using only a few available examples, thus significantly enhancing adaptability and predictive accuracy without requiring extensive retraining or large datasets.

3.5. Computational Optimization

Computational optimization techniques are incorporated into HDC-Net specifically to ensure efficient real-time inference, crucial for practical deployment in autonomous driving systems. The optimization methods employed include:

Channel-Wise Attention (Squeeze-and-Excitation): A lightweight attention mechanism recalibrates feature channel importance by adaptively scaling each channel's feature representation. Formally, each channel feature map F_c undergoes recalibration as:

$$s_c = \sigma\left(W_2 \cdot ReLU(W_1 \cdot z_c)\right) \tag{15}$$

where, z_c: Global average pooled representation of feature channel c,W_1, W_2: Fully connected layer weights, $\sigma(\cdot)$: Sigmoid activation function, s_c: Scaling factor applied to channel c.

This method enhances computational efficiency by emphasizing informative channels while suppressing redundant or insignificant ones, thereby reducing unnecessary computations.

Weight Quantization: Model weights are quantized from floating-point precision to lower-precision integer values (e.g., 8-bit integers):

$$\widetilde{w} = Quantize(w, 8 - bit) \tag{18}$$

where, w: Original floating-point weight,$\widetilde{w}$: Quantized integer weight.

Quantization significantly reduces memory usage and computational load, directly contributing to faster inference and lower resource consumption during model deployment.

Kernel Fusion: Sequential convolutional, batch normalization, and ReLU activation operations are fused into a single computational step:

$$Y = ReLU\left(BatchNorm\big(Conv(X)\big)\right) \rightarrow Y = FusedConv(X) \tag{16}$$

where, X: Input feature tensor, Y: Output feature tensor after fusion, FusedConv$(\cdot)$: Single optimized fused operation.

Kernel fusion minimizes intermediate memory access overhead, reducing inference latency and enabling rapid processing speed essential for real-time collision prediction. Collectively, these optimization strategies ensure that HDC-Net maintains efficient real-time processing capabilities suitable for practical, on-vehicle autonomous driving environments.

4. EXPERIMENTAL STUDY

This section checks how HDC-Net works when tested on real accident scenes. The DeepAccident dataset, which includes many driving events and sensor inputs from different angles, is used for the test. Crash and non-crash samples are included, such as a time when a bus suddenly turned near the market. To keep fairness, data is split into four parts and used in rotation for testing and training. The model is checked for four things—if crash can be found, how early it gives warning, how close the guessed path is to the real one, and how quickly it works. With IChOA-CNN-LSTM [8] and BCSSN [7], HDC-Net is compared, which are two other models. Using a high-speed GPU system with fixed setup, all testing is done to avoid performance change due to machine. Results show how models behave differently under load. In many crash types like bike meeting at corner, HDC-Net gives better timing and fewer mistakes. More time is taken or path guess misses by bigger value in some other models. When dataset becomes large, accuracy drops or speed reduces more in few models. This section shows full view of model performance and where struggle is faced in road conditions.

4.1. Dataset

Experiments are done using the DeepAccident dataset [18], which contains around 57,000 frames and 285,000 labeled samples from 691 driving cases. Different road types like highway, city, and village scenes are included in the data, recorded under changing weather and light. The dataset covers 12 crash types such as pedestrian and rear-end events, and also includes normal driving and near-miss situations. A 32-beam LiDAR sensor operating at 10 Hz and six RGB cameras with 360° view are used in each case.

Fixed-size inputs are formed after aligning the LiDAR and camera data in time and space. Sensor readings are scaled to a common range after alignment. To add variation, artificial weather, light changes, and noise are applied to the sensor data. Each fold used for testing is rotated once while the remaining three folds are used for model training.

With CUDA, an NVIDIA RTX 3090 GPU is used to train the model. A batch size of 32 and learning rate of 0.001 are set using the Adam optimizer. Until the training loss stabilizes, training continues for multiple rounds. The setup used to train and evaluate all models is kept the same to allow fair comparison.

Two earlier models are selected to compare with HDC-Net. IChOA-CNN-LSTM [8] extracts image features using CNN and uses LSTM for learning motion, along with a chimp-based optimization method for feature combination. A two-way LSTM and CNN are used in BCSSN [7] to process spatial and motion data together. From multiple cameras and LiDAR, bird's-eye-view images are created and used as input in BCSSN.

Sensor values are kept within the same range by adjusting their numbers. Real road feel is given by adding small changes like bad road, light change, and weather shift. These steps are used to prepare model for unknown road scenes the Table 1 summarizes the key attributes.

Table 1. Dataset statistics.

Attribute	Value
Total Scenarios	691
Total Frames	57,000
Total Annotated Samples	285,000
Sensor Types	RGB Cameras, LiDAR
Camera Views per Vehicle	6 (360° coverage)
LiDAR Configuration	32-beam
Recording Frequency	10 Hz
Collision Types	12 types (rear-end, side impact, pedestrian, etc.)
Road Conditions	Urban, highway, rural, signalized, unsignalized
Weather Variations	Day, night, rain, fog

4.2. Evaluation Metrics

Performance is measured using four main metrics. The percentage of correctly predicted collision events is shown by Collision Prediction Accuracy (CPA); better prediction is suggested by higher values. Time-to-Collision Error (TCE) calculates the mean absolute difference between predicted and actual collision times in seconds; better timing is indicated by lower values. The average distance in meters between real and predicted object paths is measured by Trajectory Deviation (TD); more accurate path prediction is reflected by smaller values, Abalation study. Computational Efficiency (CE) reports inference time per frame in milliseconds on average; faster processing is shown by lower time. For reference, precision, recall, F1-score, and AUC-ROC are also noted.

4.3. Experimental Setup

The model is trained using Adam optimizer with a learning rate of 0.001 and a batch size of 32. The training process runs for multiple cycles, adjusting weights based on errors. Loss minimization directs the parameter updates. The batch size balances computational load and gradient stability.

All experiments run on an NVIDIA RTX 3090 GPU with CUDA acceleration. The setup allows high-speed matrix operations and large dataset processing. Training and testing are performed under the same configuration to ensure consistency.

A 4-fold cross-validation strategy divides the dataset into four subsets. Three subsets train the model, while one tests its performance. This process repeats four times, ensuring every subset serves as a test set once. The rotation improves adaptability across different accident cases and traffic conditions.

The model undergoes regression testing to track stability in trajectory forecasting and collision detection. Each version is compared to previous outputs. Any deviation in motion prediction or risk assessment is flagged for further evaluation. The test ensures updates do not alter expected behavior or introduce inconsistencies.

4.4. Results and Analysis

This section presents the comparative performance evaluation of HDC-Net against contemporary models, followed by detailed ablation analyses and scalability assessments. The performance comparison focuses on key metrics including Collision Prediction Accuracy (CPA), Time-to-Collision Error (TCE), Trajectory Deviation (TD), and Computational Efficiency (CE). Table 2 summarizes these metrics for HDC-Net, IChOA-CNN-LSTM, and BCSSN.

Table 2. Performance comparison of HDC-Net and baseline models.

Model	CPA [%] ↑	TCE [s] ↓	TD [m] ↓	CE [ms/frame] ↓	Precision [%] ↑	Recall [%] ↑	F1-Score ↑	AUC-ROC ↑
HDC-Net	89.3	0.42	0.17	18.4	91.2	90.1	90.6	0.94
IChOA-CNN-LSTM	84.7	0.56	0.29	23.1	86.5	84.9	85.7	0.89
BCSSN	78.9	0.73	0.41	15.2	79.1	78.3	78.7	0.85

Fig. 2 HDC-Net achieves the highest collision prediction accuracy of 89.3%, surpassing IChOA-CNN-LSTM (84.7%) and BCSSN (78.9%) by margins of approximately 4.6% and 10.4%, respectively. A significantly lower time-to-collision error (0.42 seconds) indicates that HDC-Net provides more timely and accurate collision warnings compared to the other models. Furthermore, HDC-Net demonstrates superior trajectory accuracy, with an average deviation of only 0.17 meters. Although BCSSN exhibits lower computational time per frame (15.2 ms/frame), its significantly lower accuracy suggests a critical performance trade-off, whereas HDC-Net maintains high predictive reliability while still achieving real-time inference (18.4 ms/frame).

Fig. 2. Comparative graphs of CPA, TCE, TD, and CE metrics across models.

Ablation Study: To evaluate the individual contributions of key components within HDC-Net, an ablation analysis was conducted, systematically removing each critical component (GAN-based data augmentation, dual-branch RNN encoder, self-attention, and MAML-based meta-learning) to observe their impacts. Table 3 summarizes these results.

Table 3: Ablation study results for HDC-Net.

Model Variant	CPA [%] ↓	TCE [s] ↑	TD [m] ↑	CE [ms/frame] ↓
HDC-Net (Full model)	89.3	0.42	0.17	18.4
Without GAN augmentation	86.1	0.51	0.23	17.9
Single-branch (LSTM only)	85.4	0.53	0.25	17.5
Single-branch (GRU only)	84.8	0.54	0.27	17.4
Without self-attention	84.2	0.57	0.30	17.8
Without meta-learning (MAML)	83.7	0.58	0.32	17.6

Removing the GAN-generated synthetic data causes a notable drop in accuracy (from 89.3% to 86.1%), underscoring the importance of augmented collision scenarios for addressing data scarcity. The dual-branch temporal encoder demonstrates clear value; using only LSTM or GRU individually results in decreased accuracy (down to 85.4% and 84.8%, respectively), reflecting the complementary nature of multi-scale temporal modeling as shown in Fig. 3.

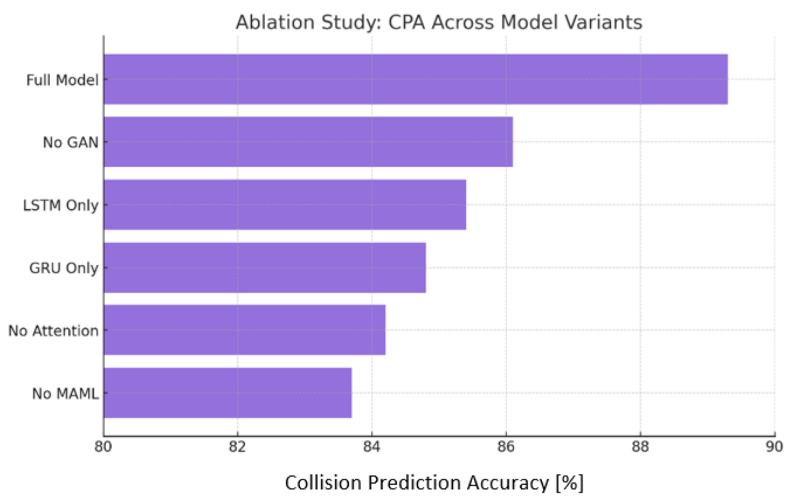

Fig. 3. Ablation study results graph highlighting CPA changes across variants.

The self-attention mechanism is similarly essential, as its removal raises time-to-collision error and trajectory deviation, confirming its critical role in filtering sensor noise. The absence of MAML-based meta-learning exhibits the most substantial reduction in performance on previously unseen scenarios, with accuracy decreasing significantly to 83.7%. This explicitly validates the necessity of meta-learning for rapid model adaptation.

Scalability Analysis: The scalability of HDC-Net and baseline models was evaluated under increasing dataset sizes. The dataset scaling levels are defined explicitly as follows: "1x" refers to the original dataset size, "2x" to a doubled dataset size, and "3x" to a tripled dataset size. Table 4 presents the results of these experiments.

Table 4: Scalability and performance under increasing dataset sizes.

Model	CPA [%] (1x)	CPA [%] (2x)	CPA [%] (3x)	Processing Time Increase [%]
HDC-Net	89.3	88.7	87.9	12.1
IChOA-CNN-LSTM	84.7	83.1	81.5	21.3
BCSSN	78.9	76.4	73.2	8.4

As shown in Table 4, HDC-Net maintains superior performance under increasing dataset sizes, demonstrating only a marginal drop (1.4%) from the original to the tripled dataset scenario (89.3% to 87.9%). In contrast, IChOA-CNN-LSTM experiences a more substantial decline of 3.2%, and BCSSN declines even further (5.7%). The notable stability of HDC-Net is primarily due to the integrated meta-learning approach, which significantly enhances adaptability and generalization with growing data as shown in Fig. 4.

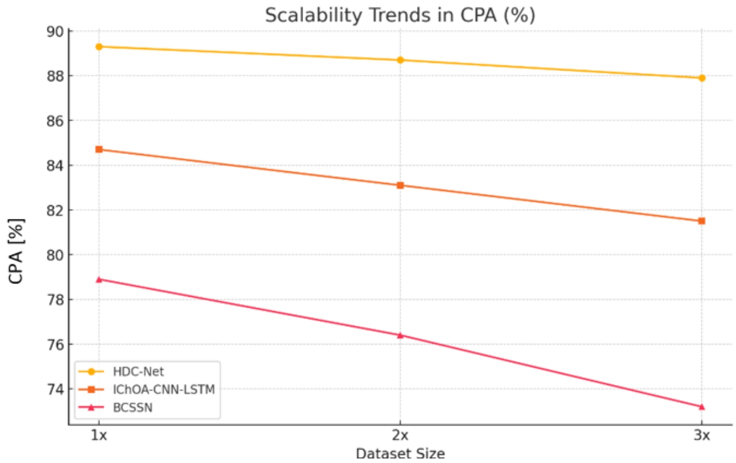

Fig. 4. Scalability trends in CPA [%].

Processing time increases moderately for HDC-Net (12.1%), illustrating its robust real-time optimization as shown in Fig. 5. Conversely, IChOA-CNN-LSTM, despite its higher complexity, shows the largest increase in processing time (21.3%), reflecting efficiency constraints under scale. BCSSN maintains minimal time increase (8.4%) due to its simpler model architecture, but its accuracy is considerably compromised, making it less reliable overall. BCSSN has the lowest processing time increase, reflecting its simpler structure, but at a significant cost to predictive accuracy. HDC-Net and IChOA-CNN-LSTM show moderate and higher increases, respectively, but HDC-Net retains a favorable balance between computational demand and predictive performance.

In summary, the comprehensive results demonstrate that HDC-Net significantly outperforms contemporary models (IChOA-CNN-LSTM and BCSSN) in collision prediction accuracy, temporal precision, trajectory fidelity, and real-time computational efficiency. The ablation study explicitly confirms the essential role of the dual-branch temporal encoder, GAN augmentation, self-attention module, and meta-learning adaptation, establishing their clear individual and collective contributions. Scalability experiments further reinforce HDC-Net's robust adaptability to larger and varied datasets, highlighting the effective integration of meta-learning and real-time optimizations. Thus, the proposed HDC-Net provides a cohesive and effective solution to real-time collision prediction challenges faced by autonomous driving systems.

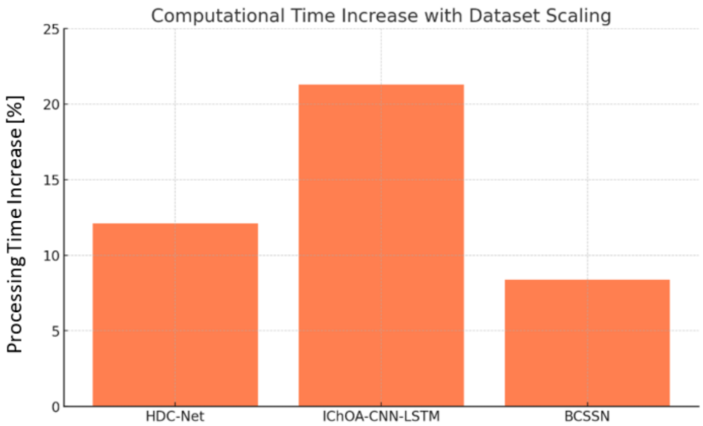

Fig. 5. Processing time increase with dataset scaling.

4.5. Discussion

The discussion summarizes key findings from the study. HDC-Net achieves higher collision prediction accuracy (89.3%) than IChOA-CNN-LSTM and BCSSN. The proposed model provides accurate predictions within 0.42 seconds and maintains trajectory deviations around 0.17 meters. Although BCSSN predicts collisions faster (15.2 ms/frame), it has lower accuracy. HDC-Net balances accuracy with moderate computational demands (18.4 ms/frame), largely due to optimizations like attention pooling and kernel fusion. Such optimization helps achieve acceptable real-time prediction speed, important for practical vehicle deployment.

Adaptability to new conditions varies among evaluated models. HDC-Net's accuracy generally stays stable (only 1.4% drop) as the dataset size triples, benefiting from meta-learning and GAN-generated synthetic data. Conversely, IChOA-CNN-LSTM loses around 3.2%, and BCSSN accuracy declines roughly 5.7% with more data. Ablation results confirm that the meta-learning approach plays a major role in maintaining performance when conditions change. Without meta-learning, accuracy drops to about 83.7%, which suggests that rapid adaptation is essential for maintaining prediction reliability. GAN-generated data also helps the model handle rare collision events, although how realistic these generated events are remains unclear. Thus, broader validation of GAN-generated scenarios might improve confidence in these results.

Real-time prediction remains a challenge for vehicle-based models. While HDC-Net runs efficiently on high-performance GPUs, deployment on embedded hardware used in cars could be more difficult due to complexity. Future research could investigate simpler versions of the model, perhaps using techniques like pruning or quantization, to better meet the limits of automotive hardware. Furthermore, this study does not test real-time performance in actual vehicles or simulation environments. Such practical testing is essential to determine if the current implementation can realistically provide timely collision warnings under normal driving conditions. Future work should focus on real-world simulations or hardware-in-the-loop evaluations to confirm practical feasibility.

This study's evaluations were performed using only the DeepAccident dataset. While DeepAccident provides diverse scenarios, relying on a single dataset may limit generalizability. Testing on additional datasets or real driving scenarios could reveal more about HDC-Net's general applicability. Also, using synthetic GAN-generated data could introduce unrealistic elements or subtle biases, since no quantitative method assessed data realism. Addressing this issue might involve validating synthetic data with quantitative realism measures or comparing predictions from real and synthetic scenarios.

Analyzing error patterns provides insights into model limitations. HDC-Net occasionally misses collisions that happen suddenly, suggesting challenges predicting events with minimal warning signals. False positives occur generally in dense or complicated driving environments, perhaps caused by sensor noise or sudden vehicle movements the model interprets incorrectly. The attention mechanism partly reduces these issues by filtering irrelevant information but cannot fully prevent misclassification. Another observed weakness is prediction uncertainty during uncommon driving conditions, such as heavy rain or nighttime driving, which lack sufficient training examples. Expanding training data to include more examples of these challenging situations could reduce uncertainty.

In summary, the discussion highlights several strengths and limitations of HDC-Net. The model successfully combines spatial-temporal modeling, noise filtering, meta-learning, and synthetic data augmentation, addressing important gaps found in earlier approaches. However, several limitations remain, including dataset generalizability, synthetic data realism, and computational demands for vehicle deployment. Future research should address these limitations, focusing on validation across additional scenarios, improving data realism, and testing more practical model variants. Such steps could improve reliability and practical usefulness of collision prediction systems in autonomous vehicles.

5. CONCLUSION

This paper presented the Hybrid Deep Collision Prediction Network (HDC-Net), a unified deep-learning framework designed for real-time collision prediction in autonomous driving environments. HDC-Net integrates a dilated convolutional neural network (CNN) with a dual-branch recurrent neural network (combining LSTM and GRU) to capture comprehensive spatial and temporal features. It further employs a self-attention mechanism to effectively filter sensor noise, utilizes Model-Agnostic Meta-Learning (MAML) for rapid adaptation to new scenarios, and addresses data scarcity by augmenting collision data through a generative adversarial network (GAN). Evaluated on the DeepAccident dataset using rigorous 4-fold cross-validation, HDC-Net achieved a collision prediction accuracy of 89.3%, a mean time-to-collision error of 0.42 seconds, and an average trajectory deviation of 0.17 meters, while maintaining real-time inference at approximately 18.4 ms per frame. These results represent notable improvements over state-of-the-art baseline models, outperforming them by margins of approximately 4.6% and 10.4%. The integration of dual-scale temporal modeling, attention-driven noise reduction, meta-learning-based fast adaptation, and synthetic data augmentation collectively enables HDC-Net to address critical limitations of prior collision prediction models, notably in robustness, generalization to new conditions, and real-time viability. Nevertheless, the study is constrained by evaluation on a single dataset and lacks validation under actual on-road driving conditions. Future research directions include evaluating the model's generalization capability across multiple diverse datasets, real-world or simulated environments, and exploring advanced model compression techniques to develop more lightweight variants suitable for deployment on embedded automotive hardware. Overall, HDC-Net provides a robust and adaptive framework for enhancing collision prediction accuracy and reliability in autonomous driving systems, balancing computational efficiency with predictive performance, and laying a clear foundation for future advancements in vehicular safety.

REFERENCES

[1] F. Chan, F. Hsiang, Y. Chen, Y. Xiang, M. Sun, "Anticipating accidents in dashcam videos," Computer Vision – ACCV, 2017, doi: 10.1007/978-3-319-54190-7_9.

[2] L, Antonio, A. Maqueda, C. del-Blanco, D. Scaramuzza, "DroNet: learning to fly by driving," *IEEE Robotics and Automation Letters,* vol. 3, no. 2, pp. 1088–1095, 2018, doi: 10.1109/lra.2018.2795643.

[3] J, Vargas, A. Suleiman, T. Onur, R. Rahul, J. Santos, "An overview of autonomous vehicles sensors and their vulnerability to weather conditions," *Sensors*, vol. 21, no. 16, p. 5397, 2021, doi: 10.3390/s21165397.

[4] R. Adrian, K. Faragó, "Collision avoidance using deep learning-based monocular vision," *SN Computer Science*, vol. 2, no. 5, 2021, doi: 10.1007/s42979-021-00759-6.

[5] F. Chelsea, P. Abbeel, S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," International Conference On Machine Learning, 2017, doi: 10.48550/arXiv.1703.03400.

[6] L. Raphael, "Generative adversarial nets," *Machine Learning under Malware Attack*, pp. 73–76, 2023, doi: 10.1007/978-3-658-40442-0_9.

[7] L. Haichuan, L. Zhou, A. Knoll, ''BCSSN: Bi-Direction Compact Spatial Separable Network for Collision Avoidance in Autonomous Driving,'' *Robotics*, 2023, doi: 10.48550/arXiv.2303.06714.

[8] R. Geethanjali, A. Valarmathi, "A novel hybrid deep learning IChOA-CNN-LSTM model for modality-enriched and multilingual emotion recognition in social media," *Scientific Reports*, vol. 14, no. 1, 2024, doi: 10.1038/s41598-024-73452-2.

[9] X. Gao, C. Long, T. Kangsheng, C. Yingfeng, W. Hai, L. Yicheng, "Deep learning-based hybrid model for the behaviour prediction of surrounding vehicles over long-time periods," *IET Intelligent Transport Systems*, vol. 16, no. 10, pp. 1404–1412, 2022, doi: 10.1049/itr2.12214.

[10] Y. Xu, P. Quan, W. Zengfu, H. Baoquan, "A novel trajectory prediction method based on CNN, BiLSTM, and Multi-Head attention mechanism," *Aerospace*, vol. 11, no. 10, p. 822, 2024, doi: 10.3390/aerospace11100822.

[11] J. Zhao, Y. Wenchen, Z. Feng, "A CNN-LSTM-attention model for near-crash event identification on mountainous roads," *Applied Sciences*, vol. 14, no. 11, p. 4934, 2024, doi: 10.3390/app14114934.

[12] A. Vaibhav, Y. Shih, H. Brandon, M. Deepan, K. Pramod, M. Al Faruque, "Spatiotemporal scene-graph embedding for autonomous vehicle collision prediction," *IEEE Internet of Things Journal*, vol. 9, no. 12, pp. 9379–9388, 2022, doi: 10.1109/jiot.2022.3141044.

[13] A. Pawar, "Modified model-agnostic meta-learning," IEEE International Conference on Machine Learning and Applied Network Technologies, 2020, doi: 10.1109/icmlant50963.2020.9355973.

[14] L. Zhang, Z. Chenbo, Z. Jiajia, G. Jihong, Z. Shuigeng, "Meta-ZSDETR: zero-shot DETR with meta-learning," CVF International Conference on Computer Vision, 2023, doi: 10.1109/iccv51070.2023.00630.

[15] L. Yanfen, W. Hanxiang, L. Dang, T. Nguyen, D. Han, A. Lee, I. Jang, H. Moon, "A deep learning-based hybrid framework for object detection and recognition in autonomous driving," *IEEE Access*, vol. 8, pp. 194228–194239, 2020, doi: 10.1109/access.2020.3033289.

[16] D. Liu, W. Xin, C. Yao, L. Zihao, L. Zheng, "A novel framework of real-time regional collision risk prediction based on the RNN approach," *Journal of Marine Science and Engineering*, vol. 8, no. 3, p. 224, 2020, doi: 10.3390/jmse8030224.
[17] S. Seo, J. Hoon "A robust collision prediction and detection method based on neural network for autonomous delivery robots," *ETRI Journal*, vol. 45, no. 2 pp. 329–337, 2023, doi: 10.4218/etrij.2021-0397.
[18] T. Wang, K. Sukmin, W. Ji, X. Enze, G. Chongjian, C. Junsong, L. Zhenguo, L. Ping, "DeepAccident: a motion and accident prediction benchmark for V2X autonomous driving," Conference on Artificial Intelligence, vol. 38, no. 6, pp. 5599–5606, 2024, doi: 10.1609/aaai.v38i6.28370.