Complex & Intelligent Systems
https://doi.org/10.1007/s40747-024-01463-7

ORIGINAL ARTICLE

Sound event detection in traffic scenes based on graph convolutional network to obtain multi-modal information

Yanji Jiang[1] · Dingxu Guo[1] · Lan Wang[3] · Haitao Zhang[3] · Hao Dong[2] · Youli Qiu[1] · Huiwen Zou[3]

Received: 31 January 2024 / Accepted: 17 April 2024

Abstract

Sound event detection involves identifying sound categories in audio and determining when they start and end. However, in real-life situations, sound events are usually not isolated. When one sound event occurs, there are often other related sound events that take place as co-occurrences or successive occurrences. The timing relationship of sound events can reflect their characteristics. Therefore, this paper proposes a sound event detection method for traffic scenes based on a graph convolutional network, which considers this timing relationship as a form of multimodal information. The proposed method involves using the acoustic event window method to obtain co-occurrences or successive occurrences of relationship information in the sound signal while filtering out possible noise relationship information. This information is then represented as a graphical structure. Next, the graph convolutional neural network is improved to balance relationship weights between neighbors and itself and to avoid excessive smoothing. It is used to learn the relationship information in the graph structure. Finally, the convolutional recurrent neural network is used to learn the acoustic feature information of sound events, and the relationship information of sound events is obtained by multi-modal fusion to enhance the performance of sound event detection. The experimental results show that using multi-modal information with the proposed method can effectively improve the performance of the model and enhance the perception ability of smart cars in their surrounding environment while driving.

Keywords Sound event detection · Graph convolutional networks · Label dependencies · Multi-modal

Introduction

The development of intelligent vehicles has significantly enhanced the existing transportation modes, resulting in improved transportation efficiency, safety, and convenience. The main objective of intelligent vehicles is to equip them with efficient and accurate autonomous perception capabilities that provide reliable external environmental information to the moving vehicle and help it understand all kinds of events occurring during the drive. These capabilities are crucial for the autonomous decision-making and autonomous driving of intelligent vehicles. However, the current perception technology of intelligent vehicles primarily relies on visual image recognition technology, overlooking the significance of sound as an important information source. Sound can provide effective information and has the advantage of being unaffected by light intensity and occlusion. Audio

✉ Hao Dong
eason@utcer.com

Yanji Jiang
jyjvip@126.com

Dingxu Guo
1051733739@qq.com

Lan Wang
lwang1216@stpt.edu.cn

Haitao Zhang
htzhang@stpt.edu.cn

Youli Qiu
superme32767@126.com

Huiwen Zou
77107149@qq.com

[1] School of Software, Liaoning Technical University, Huludao Campus, 188 Longwan South Street, Xingcheng, Huludao 125105, Liaoning, China

[2] Suzhou Automotive Research Institute, Tsinghua University, Building 5, 2266 Yangyang Road, Xiangcheng District, Suzhou 215200, Jiangsu, China

[3] Shantou Polytechnic, Next to Donghu Village, Dahao Guangao Street, Haojiang District, Shantou 515078, Guangdong, China

Published online: 16 May 2024

monitoring equipment is also low in cost, small in size, quick to install, not easily damaged, and simple to maintain. Therefore, identifying and detecting events in road traffic through sound signals holds great significance.

Sound event detection is predominantly based on deep learning technology. In the beginning, researchers tested traditional network structures such as CNN [1] and RNN [2] to confirm their effectiveness. Later, Cakir et al. [3] introduced the Convolutional Recurrent Neural Network (CRNN) for sound event detection. This model combines the strengths of both CNN and RNN by capturing local and temporal information, respectively. Due to its superior performance, it has become the most commonly used model architecture for sound event detection. Since then, scholars have made numerous improvements to the algorithm model and achieved successful results. For example, Lu et al. [4] replaced the RNN portion of the CRNN structure with a bidirectional GRU to take advantage of contextual information. Xia et al. [5] enhanced significant channel and time–frequency information by modeling the interdependence between the time–frequency domain and multiple channels. Watcharasupat et al. [6] combined the cross-entropy loss function with Dice loss to minimize the interference of negative samples and enhance the robustness of the model. Wang et al. [7] explored deterministic information in frame-by-frame prediction and the basic principle of frames. They designed a frame-based loss to improve the model's detection accuracy. Feroze et al. [8] reduced the error rate of a single category by using PLP features instead of Mel-frequency and loudness cepstral coefficients. Adavanne et al. [9] obtained lower error rates than single-channel features by extracting dual-channel features in binaural audio.

It's worth noting that most of the studies conducted on sound event detection have only considered the acoustic features of the sound. As a result, the models lack the prior knowledge and experience that humans possess when it comes to identifying different sound events, which is essential to improving the performance of task-oriented feature spaces [10]. This means that the models can only analyze independent acoustic features, which can be influenced by multiple sound events, especially when they overlap. The human auditory system is highly efficient, and our brains make use of previously learned knowledge and experience to process different sounds. While current research has achieved close to or even better than human accuracy in detecting single sound events, there are still gaps in identifying complex and diverse polyphonic sound events, such as those found in driving environments. One of the main reasons is the lack of prior knowledge about the multimodal information of sound events.

In recent years, researchers have been focusing on acquiring and using multimodal information on sound events, and significant progress has been made. Tonami et al. [11] used multi-task learning, combining sound event detection and acoustic scene detection methods, to analyze sound events and acoustic scenes jointly, which improved the detection performance of both. However, the addition of acoustic scene detection tasks increased the final model parameters, making it cumbersome to deploy and apply to intelligent vehicles. Komatsu et al. [12] proposed a sound event detection method based on the probability chain rule. It performs binary detection on each event one by one and effectively improves the detection performance. However, its performance depends on the sequence of sound event classes set in advance, and it is not stable enough in complex and changeable driving environments.

To extract sound event relationship information from real sound data sets, we can mine the statistical relationship of label information in the scene. This statistical information is obtained from a large number of data points and is more realistic and robust than other multi-modal information obtained from artificial settings. According to this idea, papers [13–15] have conducted related research. They extracted the co-occurrence relationship of sound events from label information and used a graph neural network for learning to apply it to the task of sound event classification. However, these studies mainly focus on the co-occurrence relationship of sound events and fail to extract the relationship between sound events more comprehensively, and to some extent, the graph sparsity problem occurs, which reduces the efficiency of information transmission. For this, shallow graph neural networks may not capture enough graph structure information, and too-deep networks may lead to excessive smoothing, which makes the model easy to overfit. It increases the complexity and may introduce feature noise, which affects the performance of the model [16]. At the same time, the current research mainly focuses on sound event classification, and insufficient attention is paid to the task of sound event detection.

Our proposed approach, called sound event window, uses a relation extraction method to extract the temporal relation information of sound events in audio datasets. This includes co-occurrence relations and successive occurrence relations. To simulate human knowledge in this respect, we use a graph convolutional network to learn the relationship between events. This is then multimodally fused with the acoustic feature information extracted by the convolutional recurrent neural network to improve the model's detection ability.

The paper presents three main contributions:

1. A sound event window is proposed to improve the extraction of sound event relationships. This makes it possible to obtain information on the relationship between sound events that co-occurrence and successive occurrence,

leading to a more comprehensive understanding of event relationships.
2. A more comprehensive event relationship is used to construct the graph, which addresses the issue of too sparse graph structure when using a co-occurrence relationship to construct the graph.
3. The graph neural network is introduced in the sound event detection task, where relationship information between events is obtained through multimodal fusion. Comprehensive comparative experiments and ablation experiments are conducted, and a variety of evaluation indicators are used to measure the model's performance.

Related work

Graph neural networks have gained attention due to their exceptional performance in recent years. Researchers have started exploring the use of graph neural networks in multi-label image recognition and sound event classification. Wang et al. [13] developed a model that represents a graph using statistical event co-occurrence relationships in audio. They then combined the node representation acquired through GCN with the acoustic representation obtained through CNN. Sun et al. [14] learned sound event co-occurrence information through two GCNs, resulting in improved sound event classification performance. The use of graph neural networks to learn relationship information has proven to be effective in enhancing the performance of sound event classification tasks.

In their study on sound event relationships, Imoto et al. [15] used the frequency of two sound events occurring together to create an edge graph. While this method does consider the co-occurrence of sound events to a certain extent, it fails to consider the conditional probability of one event occurring when another event has already occurred. Chen [17] first addressed this problem in multi-label image recognition by counting label co-occurrence and using conditional probability to calculate the likelihood of one label co-occurring with another. Building on this method, Wang et al. [13] and Sun et al. [14] calculated the conditional probabilities of different sound events co-occurring. However, Sun et al. and Chen [17] used binary classification to create edges based on a threshold of 1, which can lead to over-smoothing during the training of graph neural networks. Furthermore, previous studies [13–15] only considered the co-occurrence of sound events without taking into account the importance of successive occurrences and causal relationships between related sound events. Counting only co-occurrence relationships to construct graphs also leads to graphs with many nodes and low connection density, which can result in low information transmission efficiency.

In comparison to other methods, the sound event window method proposed in this paper extracts more comprehensive information on the relationship between sound events by co-occurrence and successive occurrence. This enables the constructed graph structure to have more accurate information and effectively resolves the problem of graph sparsity and excessive smoothing that exists in the co-occurrence method. Moreover, unlike multi-label image recognition and sound event classification tasks, sound event detection requires the determination of the start and end time of each sound event in the audio. By extracting the event successive occurrence relationship in audio, the model can obtain more temporal information, which helps it determine the start and end time of the event.

Proposed method

The introduction of a graph neural network to the task of sound event detection was inspired by previous research on sound event classification. Sound event detection, unlike sound event classification, requires the additional detection of the start and end events for each sound event. To address this requirement, a sound event window method is proposed in this paper, aiming to capture a more comprehensive understanding of the temporal dynamics of sound events. Additionally, a G-CRNN model is designed to effectively integrate acoustic features and event relations, leveraging the power of multimodal information to enhance the performance of sound event detection.

Overall architecture

In this section, we present an architecture for sound event detection that combines graph convolutional networks (GCN) and convolutional recurrent neural networks (CRNN). The GCN component is used for learning the relationship between different sound events, while the CRNN component is used for learning the acoustic features of sound events. The two sets of features are then fused to enable sound event detection.

Below, we provide a detailed overview of the G-CRNN architecture. As shown in Fig. 1, the system consists of two main modules: the event relation learning module and the acoustic feature learning module. We describe each module in turn.

The G-CRNN model consists of two main modules and a classification network. The first module is the acoustic feature learning module, which uses a CRNN to extract acoustic features and timing information from sound events. The second module, the event relation learning module, builds a graphical structure based on data labels and uses a two-layer GCN to learn the relation information between sound events.

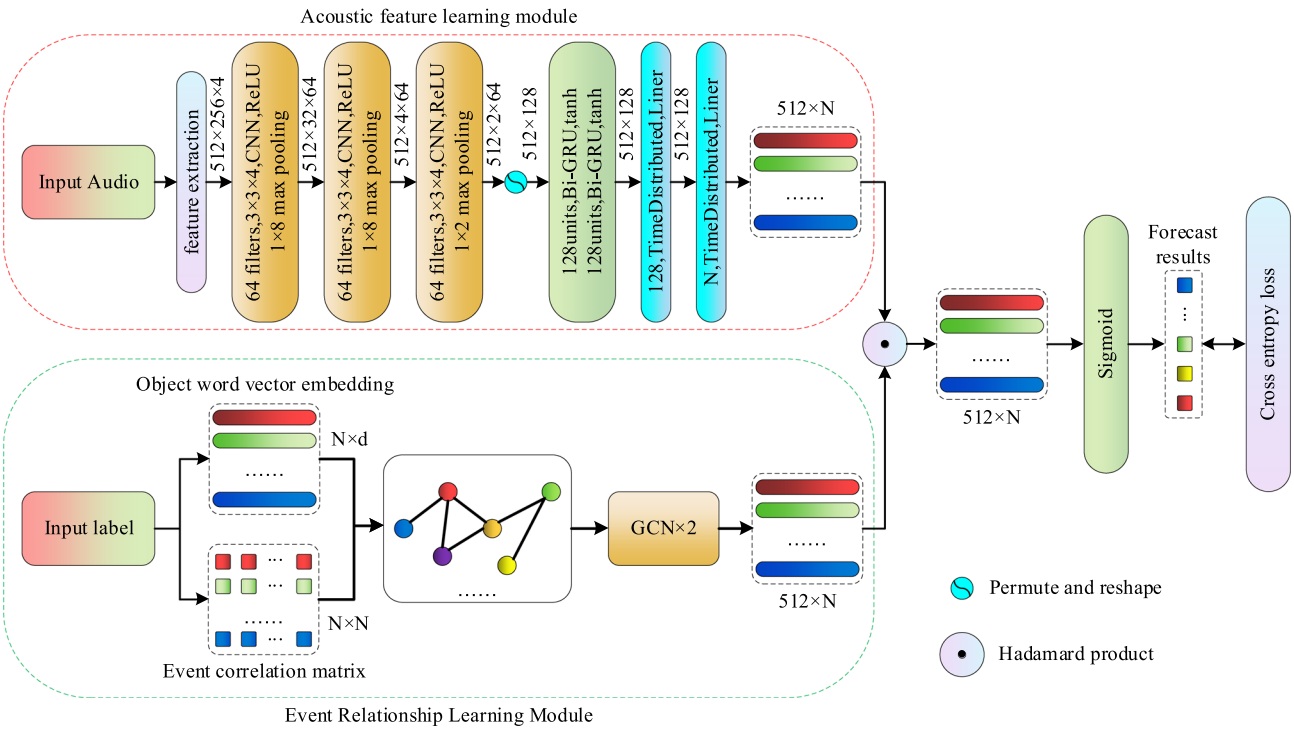

Fig. 1 The overall architecture of G-CRNN

The two modules provide multi-modal information that is fused and fed into the sound event classification network for final sound event detection.

Acoustic feature learning module

In this paper, to simultaneously capture the spectral and temporal information of sound events in audio, a convolutional recurrent neural network (CRNN) is selected as the acoustic feature learning module. The CRNN structure combines the advantages of convolutional neural networks (CNN) in capturing local features with the ability of recurrent neural networks (RNN) to process time series information, which effectively extracts the acoustic features required by sound events. In the process of building the acoustic feature learning module, this paper refers to the previous research results on sound event detection and classification tasks [1–15], based on which the model of this paper is constructed. Figure 1 shows the structure of the acoustic feature learning module in detail.

The input of the model is the audio file and the corresponding audio label in the dataset. The audio is input to the model and preprocessed to extract features. The 512-point fast Fourier transform (FFT) is used to extract the spectrogram from each channel of the two-channel audio on a Hamming window with 512 sampling points and 50% overlap. And extract the phase and amplitude of the spectrogram (using only the positive frequencies from the extracted features, not the zeroth position). The final feature vector is composed of 512 frames of feature sequence, having dimensions of 512 × 256 × 4, with 4 representing the amplitude and phase components of two channels.

After preprocessing, the feature sequence is sent to a two-dimensional CNN consisting of three layers. Each CNN layer comprises 64 filters of 3 × 3 × 4 dimensional receptive field. To keep the length of the feature sequence unchanged, the step size and fill are set to 1. After each CNN layer, the output is normalized using batch normalization and activated using the ReLU function. Then, the dimensionality is reduced using Max pooling along the frequency axis, which is 8,8,2 in frequency after each of the three CNN layers. By using a filter kernel that spans all channels, the CNN can learn relevant features in both time and frequency dimensions within the channel. The output of the last CNN layer is 512 × 2 × 64.

After the convolutional neural network (CNN) generates a feature sequence, it is reshaped into a 512 × 128 sequence and passed to the bidirectional gated recurrent unit (GRU) layer. The GRU layer has 128 nodes in each layer, and the tanh function is used for activation. The GRU layer learns the temporal context information of sound events. The final output sequence has the same dimensions as the input sequence.

The bidirectional GRU produces a sequence of features that are then passed through two TimeDistributed layers. These layers share weights and biases at each time step, which reduces the number of model parameters while helping

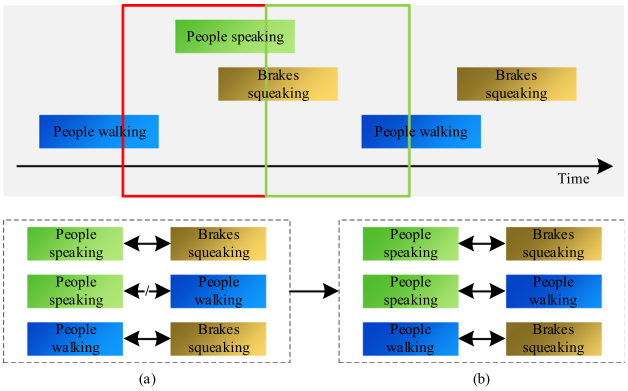

Fig. 2 Sound event window

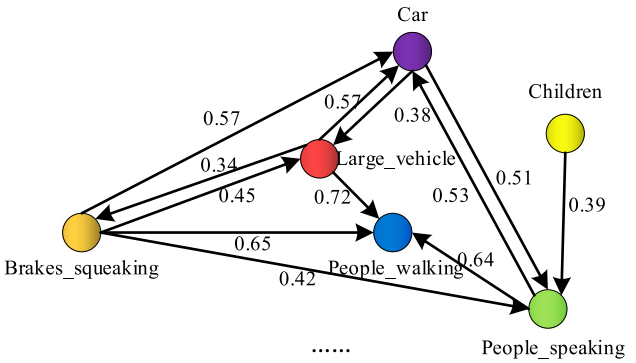

Fig. 3 Examples of partial graphs built using event relationships

the model process each sequence element more effectively. The first TimeDistributed layer consists of 128 nodes, while the second layer has N nodes. The value of N is determined by the number of categories of sound events in the corresponding dataset, both using linear activation functions. The final output sequence for acoustic features is 512 × N.

Event relationship learning module

Construction of the graph structure

To gain a more comprehensive understanding of sound event relationships, this paper proposes using a sound event window method to obtain information on both co-occurrence and successive occurrence sound events. The definition of co-occurrence and successive occurrence of sound events is as follows:

1. Co-occurrence: this refers to the traditional co-occurrence relationship, where different types of sound events overlap in duration.
2. Successive occurrence: additionally, the proposed sound event window method captures relation information for multiple non-overlapping sound events within a T-second sound event window. This is illustrated in Fig. 2.

The two boxes in Fig. 2 represent two sound event windows of length T, which are used to judge the presence of sound events within T seconds and their successive occurrence relationships. In this paper, a 60-s sound event window is used to determine and extract the relationship between sound events. Figure 2a ignores the correlation between people talking and people walking because they do not co-occurrence. However, people talking and people walking are correlated, which is successfully captured in Fig. 2b using the sound event window.

To begin, this paper utilizes the labels from the training data set to identify the sound event class and its start and end times in each audio. Each sound event category is represented as L_i, while the two events happening as co-occurrences and successive occurrences in a sound event window are denoted as L_{ij}. Here, i and j refer to the sound event category numbers, with different values representing different sound event categories, and i and j cannot be equal. Afterward, the count of each sound event category is recorded as X_i, which represents the number of occurrences of event L_i. Additionally, the number of co-occurrences and successive occurrences of two events, i.e., the number of L_{ij}, is recorded as X_{ij}.

The probability of event L_j co-occurrence and successive occurrence with event L_i is calculated as P_{ij} based on several various events X_i and two events X_{ij}.

$$P_{ij} = P(L_j|L_i) = X_{ij} \Big/ X_i \tag{1}$$

It's worth noting that P_{ij} and P_{ji} are not necessarily the same, and they have different meanings. For instance, if L_i represents the sound of a violent collision from a car accident and L_j is the sound of a moving car, the probability of L_j happening as a co-occurrence of L_i is very high. However, the probability of Li happening when L_j is present is very low. Therefore, P_{ij} and P_{ji} are not equal, and they represent different meanings.

When calculating the probability of various sound events, there may be some cases where the training data is different from the inference data. This can lead to small probability relations becoming noise data during model validation and testing. To prevent this from negatively affecting the generalization of the graph, a filtering threshold α is used to remove the noisy relations with very low probability and reset their weights to 0. These thresholds are set between 0 and 1.

$$P_{ij} = \begin{cases} P_{ij}, & P_{ij} \geq \alpha \\ 0, & P_{ij} < \alpha \end{cases} \tag{2}$$

Figure 3 illustrates the graph structure built by considering different types of sound events as nodes and the calculated

	(object) banging	Bird singing	Car	Children	People speaking	People walking	Wind blowing
(object) banging	0.00	0.10	0.87	0.17	0.28	0.21	0.00
Bird singing	0.00	0.00	0.08	0.00	0.00	0.00	0.00
Car	0.00	0.58	0.00	0.00	0.00	0.00	0.00
Children	0.15	0.52	0.20	0.00	0.58	0.10	0.00
People speaking	0.00	0.47	0.19	0.18	0.00	0.19	0.00
People walking	0.00	0.60	0.58	0.00	0.35	0.00	0.00
Wind blowing	0.00	0.87	0.09	0.00	0.00	0.00	0.00

Fig. 4 Event relation adjacency matrix for the TUT Sound Events 2016 Residential area dataset

	Brakes squeaking	Car	Large vehicle	People walking	People speaking	Children
Brakes squeaking	0.00	0.67	0.00	0.49	0.48	0.64
Car	0.00	0.00	0.00	0.00	0.00	0.08
Large vehicle	0.08	0.15	0.00	0.00	0.50	0.39
People walking	0.28	0.60	0.00	0.00	0.43	0.61
People speaking	0.16	0.59	0.00	0.27	0.00	0.63
Children	0.08	0.59	0.00	0.38	0.49	0.00

Fig. 5 Event relation adjacency matrix for the TUT Sound Events 2017 street dataset

probabilities P_{ij} as the weights of the directed edges connecting these nodes. Each node in the graph represents a class of sound events, and an edge between nodes represents a relationship between two sound events. The weights of the edges are calculated using the method described above. The graph is computed from the data labels of one of the cross-validation datasets of the TUT Sound Events 2017-Street dataset (partially simplified for plotting purposes) and represents partial relationship information about sound events in street view. The constructed graph structure is depicted as the N × N event correlation matrix in Fig. 1. In this matrix, rows and columns represent sound event nodes, while the values assigned to each corresponding pair of nodes represent the weights of the directed edges between them.

The graph structures of all cross-validation sets from the TUT Sound Events 2016 residential area and TUT Sound Events 2017 street datasets were integrated into adjacency matrices, which are presented in Figs. 4 and 5 in this paper.

An analysis of Fig. 4 reveals that when a collision sound is detected, there is a higher probability of co-occurrence or successive occurrence detecting a car driving sound, while when a car driving sound is detected, the probability of co-occurrence or successive occurrence detecting a collision sound is relatively low (below the filtering threshold α, thus these edges are represented by a weight of 0 in the figure). This phenomenon aligns with common sense; that is, in traffic scenarios, when a collision sound occurs, it is often associated with a car, whereas the probability of a car colliding while driving normally is low.

Similarly, Fig. 5 also demonstrates a similar relationship: when detecting the sound of a car brakes squeaking, the co-occurrence or successive occurrence of a car driving sound has a higher probability, whereas after detecting a car driving sound, the co-occurrence or successive occurrence of the sound of a car brakes squeaking is relatively low (also below the filtering threshold α, thus these edges are represented by a weight of 0 in the figure). This also aligns with our common sense; that is, the occurrence of the sound of a car braking is usually accompanied by the occurrence of the sound of a car driving, whereas a car does not frequently brake while driving normally, leading to the sound of car brakes squeaking occurring frequently.

These observations suggest that by analyzing the co-occurrence and successive occurrence between sound events, useful graphical structures reflecting the actual relationships between sound events can be constructed, thus providing valuable prior knowledge for sound event detection tasks.

Graph convolutional neural network with weight ratio

In a graph neural network (GNN), the graph is a topological structure consisting of multiple nodes and edges that connect them. This structure is typically represented as $G = (V, E)$, where G represents the graph, V is the set of nodes (which typically represent entities), and E is the set of edges (which represent the relationships between the nodes).

Graph neural networks (GNNS) are based on the homogeneity assumption that connected nodes tend to share similar

properties, which provides additional information for aggregating features. This relational induction bias is a key factor that enables GNNS to outperform traditional neural networks (NNS) in many tasks [18]. In this paper, we employ graph convolutional neural networks (GCN), a GNN-based architecture that learns the final representation of nodes by aggregating information from neighboring nodes, thus demonstrating high efficiency when processing graph data. By combining graph signal processing and convolutional neural networks, GCN can simultaneously learn the attributes of nodes and the structure of the graph, and this method performs well in a variety of tasks, surpassing traditional neural network-based methods [19]. The propagation of a general multi-layer GCN between layers follows the following formula:

$$H^{(l+1)} = h(\widetilde{D}^{-\frac{1}{2}}\widetilde{A}\widetilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}) \tag{3}$$

$$\widetilde{A} = A + I_N \tag{4}$$

$$\widetilde{D} = \sum_j \widetilde{A}_{ij} \tag{5}$$

In the context of graph convolutional networks (GCN), the adjacency matrix of a graph is represented by A. I_N denotes the self-connection added to each node in the graph. N stands for the total number of nodes in the graph. After each node has added the self-connection relationship, we represent the updated adjacency matrix as $\widetilde{A}$, and the degree matrix as $\widetilde{D}$. The input of the GCN network in the l layer is denoted by $H^{(l)}$, and the parameters to be trained are represented by $W^{(l)}$. The activation function used in this paper is ReLU.

Oversmoothing is a common challenge in the training of multi-layer graph neural networks. It causes the nodes to become less discriminative, making them difficult to distinguish [20]. To address this issue, this paper proposes the use of the neighbor-to-self ratio parameter β to adjust the weight ratio of neighbor nodes and the targeted node. This adjustment helps to reduce the influence of excessive smoothing on the discriminability of node features and ensures that the network can capture enough relationship information to maintain the identity information of nodes and enhance the discriminability of nodes.

$$\widetilde{A\prime} = \begin{cases} \beta A_{ij} \Big/ \sum_{\substack{j=1 \\ i \neq j}}^{N} A_{ij}, \ if \ i \neq j \\ 1 - \beta, \ if \ i = j \end{cases} \tag{6}$$

The variable $\widetilde{A\prime}$ represents the adjacency matrix that has been adjusted to account for weights. The weight ratio between a node and its neighbors is determined by the value of β, which falls within the range of 0 to 1. A larger value of

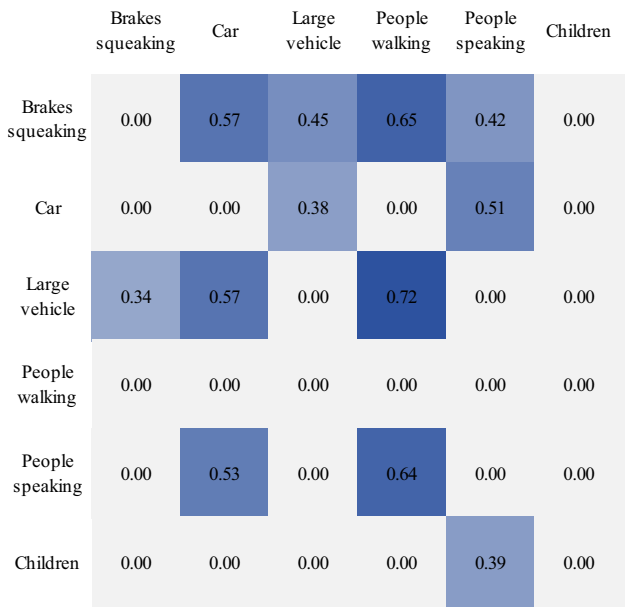

Fig. 6 Example of the adjacency matrix before adjusting the weights

β signifies that more weight is given to the neighbors, while a smaller value of β indicates that more weight is given to the node itself.

To specify the weight adjustment process, the graph structure data in Fig. 3 is taken as an example, which is first transformed into the adjacency matrix before adjusting the weights, as shown in Fig. 6. Then Formula (1) is applied to adjust the weights in the adjacency matrix, where the weight adjustment coefficient β is set to 0.3. The adjusted adjacency matrix shown in Fig. 7 is obtained.

Through weight adjustment, the weight ratio between the node and its neighbor nodes is balanced, and the degree information of the node is considered, which is the number of connections of the node in the graph structure. This adjustment not only enables the network to more accurately represent the intrinsic attributes of each sound event but also to more clearly express the relationship between events. It effectively reduces the excessive smoothing phenomenon that may occur in the process of feature learning, thereby improving the discriminability of node features. This turns the relationship matrix into a full-rank matrix. The larger the rank of the aggregation matrix, the more diversified its linear combination is. The full-rank aggregation matrix can enhance the representation ability of GNN to the greatest extent [21]. The model can more effectively identify and understand the complex interactions between sound events, thereby improving the overall detection performance.

The formula for the GCN changes after the weight ratio parameter is introduced.

$$H^{(l+1)} = h(\widetilde{D}^{-\frac{1}{2}}\widetilde{A\prime}\widetilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}) \tag{7}$$

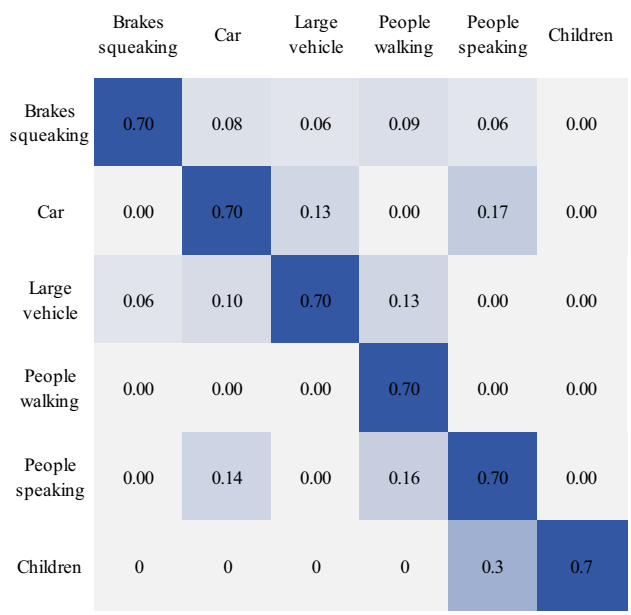

Fig. 7 Example of the adjacency matrix after adjusting the weights

To better represent the sound event and its relationship with related events, the graph neural network has been improved. This is achieved by using the neighbor-to-self ratio parameter β to balance the weight ratio between neighbor nodes and their nodes. This allows the network to fully utilize the characteristics and advantages of relationship information learning and processing.

In Fig. 1, the input node of the first GCN layer is denoted by $H^{(0)} \in \mathbf{R}^{N\times d}$. This represents the object word vector embedding of the label, where N is the number of sound event classes in the dataset and d is the dimension of the word embedding. The last layer learns the node $H^{(2)} \in \mathbf{R}^{N\times D}$, which serves as the output of the event relation learning module. Here, D is equal to the dimension of the acoustic features output by the acoustic feature learning module. The final feature sequence of the output event relation is N $\times$ 512 and is trained using the cross-entropy loss function.

To enable the merging of acoustic features with other modes, the feature sequence is converted to a size of 512 $\times$ N and saved to the document. This sequence can be obtained by training just once, and when multi-modal fusion is performed with acoustic features, the event relation feature sequence in the document can be directly accessed. This effectively reduces the training cost of the model.

Multimodal fusion and classification network

To improve the detection of sound events, this paper utilizes the Hadamard product to fuse the feature sequence of sound event relationship information and the acoustic feature sequence of sound events. The event relationship learning module outputs the former, while the acoustic feature learning module provides the latter. The fused feature sequence is 512 $\times$ N, as shown in Fig. 5. This enables the model to effectively obtain and utilize the relationship information between sound events, resulting in improved detection performance.

The fused feature sequence is fed into the sound event classification network, and the final output of the model is obtained. The network uses a sigmoid function to simultaneously activate all sound event categories to generate a prediction score between 0 and 1 for each type of sound event under each frame. These prediction scores form a frame-level prediction sequence matrix. If the score of a certain class exceeds 0.5, it is considered that this kind of sound event has been detected in this frame. By integrating the detection results of each frame, the start and end times of each kind of sound event can be determined. The network is trained using the cross-entropy loss function.

Experiment preparation

Experimental environment

The experimental equipment utilized in this research is arranged as follows: The computer operating system entails Windows 10, the CPU model employed is Intel Xeon Silver 4216, and the GPU model employed is NVIDIA Geforce RTX 3090. The implementation is carried out using the Python programming language, and the development framework is TensorFlow 2.4.0. For network training in both the event relations learning module and the acoustic feature learning module, the Adam optimizer was employed. The learning rate for the event relation learning module was set at 0.01 and the dropout rate is 0.5, while the learning rate for the acoustic feature learning module was set at 0.001 without using dropout. The maximum number of training rounds for both modules was 50 epochs. In particular, an early stopping strategy is introduced for the acoustic feature learning module to evaluate the overall model error by calculating the difference between the ER metric and the F_1 score. We stopped the training once the overall model error did not decrease for 25 consecutive training epochs.

Experimental data set

This paper makes use of the TUT Sound Events 2016 [22] and TUT Sound Events 2017 [23] datasets, which comprise genuine recordings of everyday life scenes. Specifically, these recordings capture scenes from different residential living areas and street environments. The recordings were conducted using a two-channel device with a sampling rate of 44.1 kHz and a resolution of 24 bits. Statistical information regarding the frequency of various types of sound events

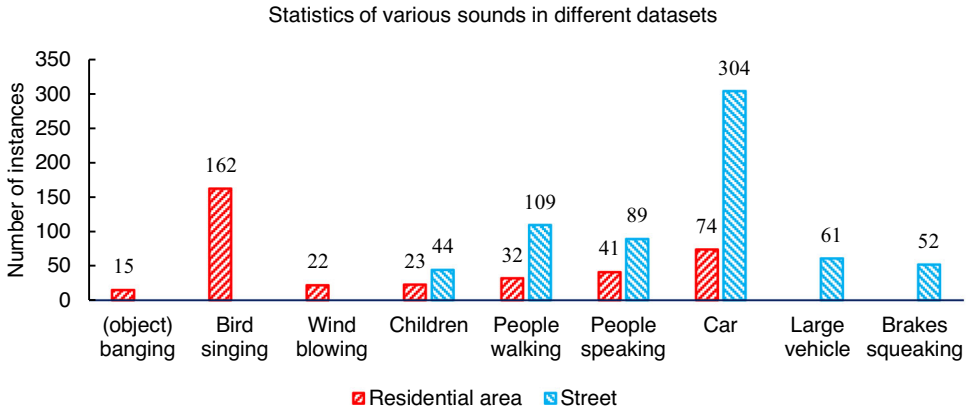

Fig. 8 Statistics of the number of instances of various types of sound events in the dataset

Table 1 Comparison of overall model performance under TUT Sound Events 2016-Residential area dataset

Model	F_1 score (%) ↑	ER metric (%) ↓	Precision (%) ↑	Recall (%) ↑
CRNN	41.42 ± 5.23	89.99 ± 1.64	49.25 ± 0.93	36.45 ± 8.31
G-CRNN(C)	48.48 ± 3.73	84.40 ± 0.92	**53.93** ± 3.89	45.37 ± 8.84
G-CRNN(CS)	**51.35** ± **2.42**	**83.65** ± **0.71**	51.41 ± **0.69**	**51.61** ± **5.36**

↑ indicates that higher is better for this index; ↓ indicates that lower is better for this index; ± is followed by standard deviation

Table 2 Comparison of overall model performance under TUT Sound Events 2017-Street dataset

Model	F_1 score (%) ↑	ER metric (%) ↓	Precision (%) ↑	Recall (%) ↑
Chen et al. [1]	30.90	85.80	–	–
Zhou et al. [2]	39.10	85.30	–	–
Cakir et al. [3]	42.66	80.84	–	–
Lu et al. [4]	39.60	82.50	–	–
Adavanne et al. [9]	41.70	79.14	–	–
Mesaros et al. [22]	42.80	93.00	–	–
Venkatesh et al.[25]	–	**75.00**	–	–
CRNN	45.62 ± 1.31	88.91 ± 6.50	45.36 ± 3.58	46.21 ± 2.28
G-CRNN(C)	46.71 ± 1.93	82.26 ± **2.47**	47.44 ± **1.75**	46.19 ± 3.64
G-CRNN(CS)	**48.62** ± **0.74**	79.60 ± 2.90	**50.37** ± 2.56	**47.12** ± **1.23**

↑ indicates that higher is better for this index; ↓ indicates that lower is better for this index; ± is followed by standard deviation; and – indicates that this index is not provided in this literature

within these two datasets is presented in Fig. 8. These two scenarios primarily revolve around vehicle driving, aligning with the requisite scenarios for assessing the sound event detection capabilities of intelligent vehicles. The development dataset was employed for training and validation purposes, while the corresponding evaluation dataset was utilized for testing and assessing the model's performance.

There are three main reasons for selecting the aforementioned datasets:

1. These two scenarios represent the primary driving scenes, aligning with the scene requirements for testing the performance of intelligent vehicle sound event detection.
2. Both datasets are derived from real recordings and contain inherent correlations between sound events in the real world. However, due to cost and other reasons, other common sound event datasets are mostly synthetic audio and lack this correlation. This hinders the ability to verify the effectiveness of introducing multi-modal information and fails to fully demonstrate the detection performance in real-life scenarios.
3. These two datasets consist of multiple segments of longer audio recordings, each containing multiple types of sound events. This allows for the exploration of potential relationships between events. In contrast, other commonly available sound event datasets often consist of

Table 3 Detection performance for each sound event in both datasets

Model	CRNN		G-CRNN(C)		G-CRNN(CS)	
Sound events	F_1 score	ER metric	F_1 score	ER metric	F_1 score	ER metric
(Object) banging	0.00	100.00	0.00	100.00	0.00	100.00
Bird singing	39.68	106.90	51.56	**102.81**	**57.25**	106.90
Brakes squeaking	**43.27**	83.43	36.21	**81.93**	28.34	84.64
Car	63.73	92.76	64.47	88.53	**65.80**	**82.98**
Children	0.52	101.14	0.00	100.00	**0.90**	102.28
Large vehicle	16.30	117.71	**28.93**	126.29	11.78	**117.43**
People speaking	**1.69**	123.94	0.74	**103.44**	0.98	108.84
People walking	17.04	109.65	16.26	**101.42**	**18.22**	101.60
Wind blowing	4.72	97.87	11.18	96.28	**19.12**	**92.50**

↑ indicates that higher is better for this index; ↓ indicates that lower is better for this index

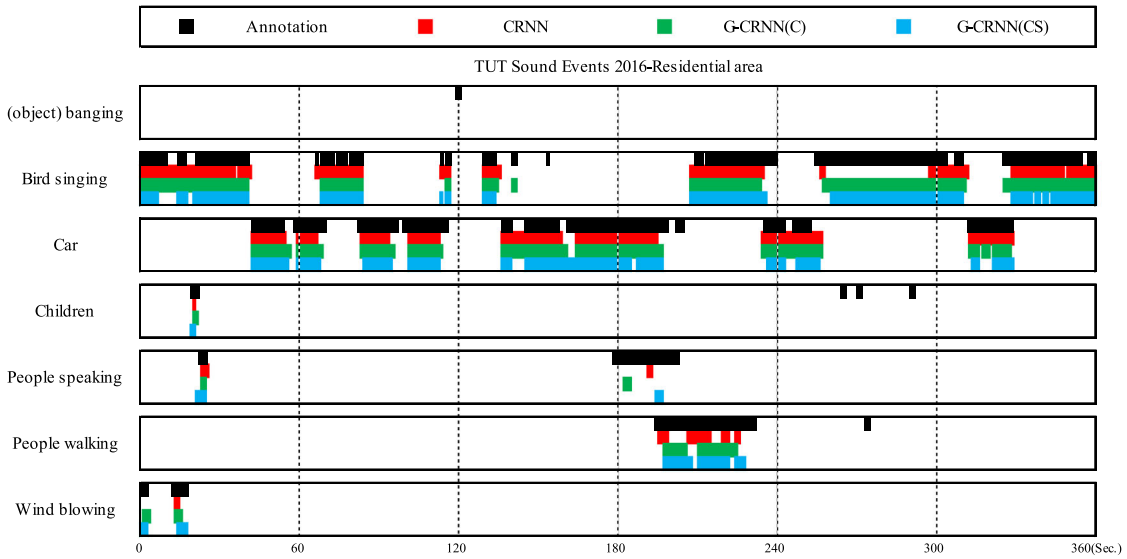

Fig. 9 Visualization of Sound event detection results for the TUT Sound Events 2016 residential area dataset

short audio clips with isolated single-sound events, which do not reflect the interrelation between events or simulate the detection requirements in real-life scenarios effectively.

1. TUT Sound Events 2016-Residential area dataset

The dataset consists of real recordings from different residential area scenarios, and the development dataset consists of 12 recordings totaling 42 min, officially divided into four sets of training and validation sets for cross-validation. The evaluation dataset consists of five recordings totaling 17 min and 49 s, which are used to test the model's performance. The dataset consists of seven independent sound event classes, and the specific event categories and number of instances are shown in Fig. 8.

2. TUT Sound Events 2017-Street dataset

The dataset consists of real recordings from different street scenes, and the development dataset consists of 24 recordings totaling 1 h, 32 min, and 8 s, officially divided into four sets of training and validation sets for cross-validation. The evaluation dataset consists of eight recordings totaling 29 min and 9 s, which are used to evaluate the model's performance. The dataset consists of six independent sound event classes, and the specific event categories and number of instances are shown in Fig. 8.

Evaluation metrics

The evaluation metrics employed in this paper encompass the F_1 score, ER measure [24], precision, and recall. These metrics are computed on non-overlapping 1-s segments. To determine the final detection performance of the model on each dataset, the mean and standard deviation of the results

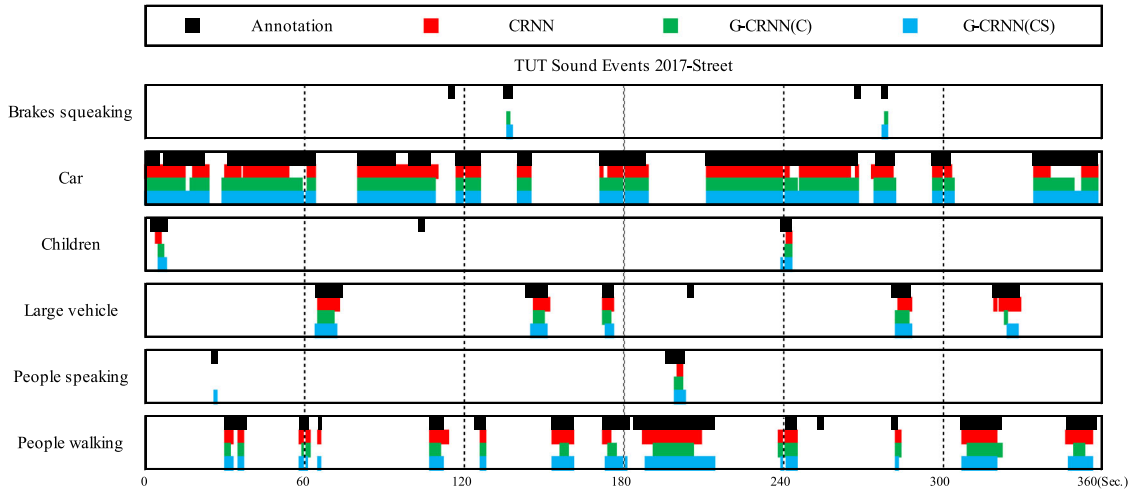

Fig. 10 Visualization of Sound event detection results for the TUT Sound Events 2017 street dataset

from all cross-validation sets are calculated. A brief description of the various evaluation metrics used in this paper is given.

The F_1 score is calculated as follows:

$$F_1 = \frac{2 \times precision \times recall}{precision + recall} \tag{8}$$

The precision and recall are calculated as follows:

$$precision = \frac{TP}{TP + FP} \tag{9}$$

$$recall = \frac{TP}{TP + FN} \tag{10}$$

The ER metric is calculated as follows:

$$ER = \frac{\sum_{k=1}^{K} S(k) + \sum_{k=1}^{K} D(k) + \sum_{k=1}^{K} I(k)}{\sum_{k=1}^{K} N(k)} \tag{11}$$

Here, k represents each 1-s segment, while $N(k)$ represents the total count of active sound event categories present in the labeled data. The variable $S(k)$ denotes the number of instances where an event is detected, yet the assigned category is incorrect. Any additional false positives and false negatives are accounted for as insertion $I(k)$ and deletion $D(k)$ errors, respectively.

Results and analysis

Overall performance experiment of the model

In this section of the experiment, the study conducted experiments on two separate datasets, comparing the CRNN model (with the event relation learning module removed while keeping other aspects consistent) against the G-CRNN (C) model, which solely utilizes co-occurrence relations, and the G-CRNN (CS) model, which incorporates both co-occurrence and successive occurrence relations. Additionally, the performance of the proposed models was compared with other methods proposed by researchers in previous studies. The three models presented in this study were trained using identical hyperparameter settings, with both the filtering threshold α and the neighbor-to-self ratio parameter β set to 0.3. The overall performance comparison of the models on both datasets is presented in Tables 1 and 2. The best values are highlighted in bold.

From Tables 1 and 2, it can be observed that incorporating the event relation module has led to an overall improvement in the performance of the models on both datasets. Specifically, the G-CRNN(C) model, which only utilizes co-occurrence relations, has shown an increase of 7.06% and 1.09% in F_1 score, a decrease of 5.59% and 6.65% in ER metric, an increase of 4.68% and 2.08% in precision, and an increase of 8.92% and a decrease of 0.02% in recall on the two datasets, respectively. This demonstrates the effectiveness of leveraging event relation learning to capture multimodal information for enhancing sound event detection performance.

Furthermore, the G-CRNN (CS) model, which incorporates both co-occurrence and successive occurrence relations, has shown additional improvements over the G-CRNN (C) model. It achieved an increase of 2.87% and 1.91% in F_1 score, a decrease of 0.75% and 2.66% in ER metric, a decrease of 2.52% and an increase of 2.93% in precision, and an increase of 6.24% and 0.93% in recall on the two datasets, respectively. These results indicate that capturing the successive occurrence relationships of sound events is beneficial for sound event detection tasks.

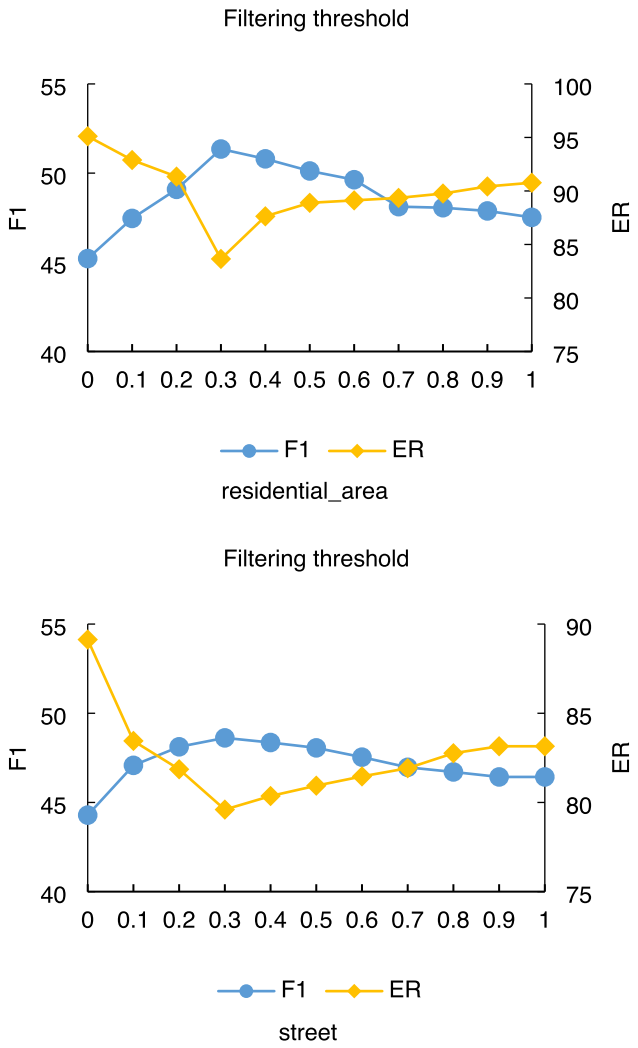

Fig. 11 Effect of filtering threshold on model performance

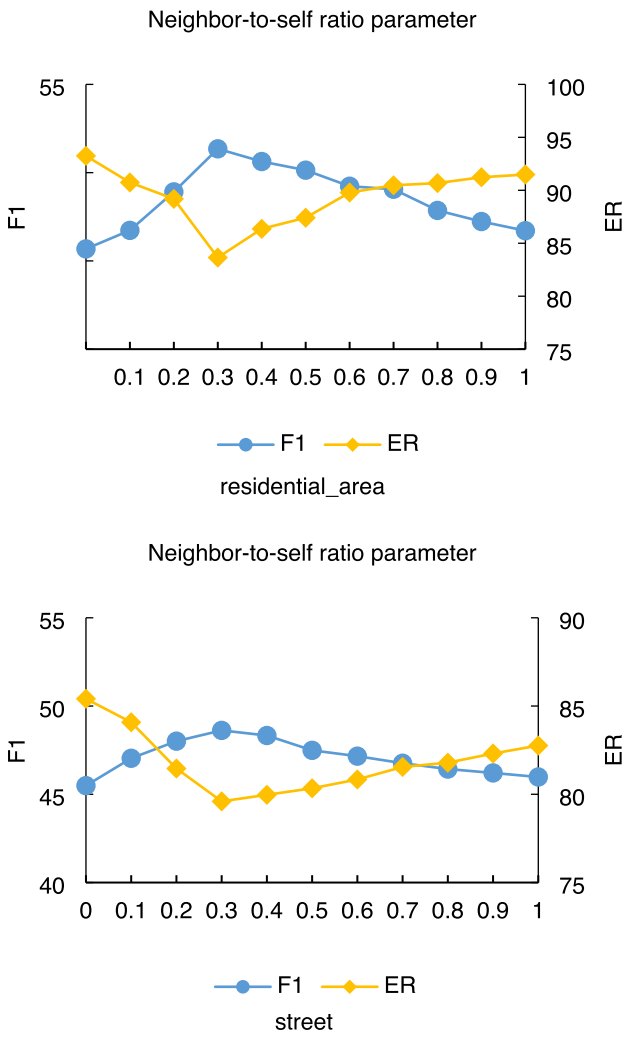

Fig. 12 Influence of adjacent self-scaling parameters on model performance

Simultaneously, the incorporation of the event relationship module enhances the stability of the detection model across multiple cross-validation sets. The standard deviation of G-CRNN (C) and G-CRNN (CS) decreases on the two datasets, with G-CRNN (CS) exhibiting a relatively lower overall standard deviation. This compellingly demonstrates that the integration of multi-modal sound event information contributes to the stability of model performance, while the inclusion of successive occurrence sound event relationships yields superior outcomes. The enhanced stability guarantees the algorithm's application performance and reasoning capabilities.

Compared to other approaches, the model demonstrates improved detection performance after incorporating multi-modal information, validating the effectiveness of the proposed method.

In comparison with other methods, the YOHO model proposed by Venkatesh et al. [25], which is similar to YOLO, achieved superior ER performance compared to the method presented in this paper, with a decrease of 4.10% in the ER metric. However, the YOHO model did not report other performance metrics and had a much larger model parameter count of 3,900,000, which is far greater than the 490,182 parameters in this paper. This high parameter count makes it challenging for deployment and application in intelligent vehicles. This paper demonstrates that by obtaining event relationships, the model's detection performance can be effectively improved even with a significantly lower parameter count, thus compensating for the disparity caused by the model's parameter count.

Performance experiments of each class of sound event detection

The experiment in this section focuses on comparing the detection performance of CRNN, G-CRNN (C), and G-CRNN (CS) on individual sound event classes using two datasets. These models align with the ones described in Experiment A. The detection capabilities of the models for all sound events in the two datasets are summarized in Table 3. The mean values of the same sound event type across the two datasets are combined, and the best performance value in the detection comparison is highlighted in bold.

Table 3 exhibits the impact of integrating the event relation module on the sound event detection performance across the two datasets. The findings demonstrate the effectiveness of leveraging multimodal information for this task. Notably, within the context of intelligent driving, significant improvements were observed for key sound event categories such as "car," "people speaking," and "people walking." The proposed G-CRNN (C) and G-CRNN (CS) models exhibited substantial reductions in ER metrics, with "Car" experiencing a decrease of 4.23% and 9.78%, respectively. Similarly, "people speaking" showed a reduction of 20.50% and 15.10%, while "people walking" exhibited a decrease of 8.23% and 8.05%, respectively. ER measures for other event categories also displayed a decrease, thus affirming the effectiveness of acquiring multimodal information in effectively mitigating false detection rates.

The primary factor contributing to the lack of improvement in the detection performance of certain sound events, such as "(object)banging," is their rarity within the driving environment. The limited size of the dataset used in this study results in a small number of occurrences (only 15) and a total duration of less than 15 s for this specific event. Consequently, the graph structure constructed for this event exhibits a scarcity of edges pointing to its corresponding node, resulting in low weights and an insufficient acquisition of relationship information. Additionally, the limited number of samples prevents the model from capturing an adequate amount of acoustic feature information, thereby hindering the accurate detection of this sound event even after feature fusion. This issue can be effectively addressed by obtaining a larger audio dataset with a sufficient number of samples.

To visually show the detection results of sound events, we make a visualization in Figs. 9 and 10, annotating the prediction results of sound events. These results show that the introduction of sound event relations can significantly improve the accuracy of sound event detection compared with the baseline method, CRNN. In particular, the G-CRNN (CS) model not only has a higher detection performance and higher detection rate but also a lower false detection rate, which fully proves the positive impact of obtaining more comprehensive information about sound event relations on detection performance.

Experiment with the influence of hyperparameter values on model performance

In this experimental phase, we investigate the impact of hyperparameter selection on model performance by conducting experiments on G-CRNN (CS) on two datasets. The filtering threshold α and the neighbor-to-self ratio parameter β are fine-tuned within the range of 0 to 1, with increments of 0.1. The results of these experiments on the two datasets are presented in Figs. 11 and 12.

Figure 11 illustrates that the model achieves the highest F_1 score and the lowest ER measure when the filtering threshold α is set to 0.3. This threshold effectively filters out noise data while preserving essential relationship information. If α is set too small, the model becomes disturbed by a significant amount of noise data, resulting in a decline in accuracy. Conversely, if α is set too large, a substantial amount of relationship information between sound events is filtered out, leading to insufficient event relationship information and a decline in model performance.

As depicted in Fig. 12, the model achieves the highest F_1 score and the lowest ER metric when β is set to 0.3. This balance allows for the acquisition of sufficient relationship information while still preserving an adequate amount of self-information.

When β is too small, the importance of neighboring information is disregarded, resulting in an inability to capture the necessary relationship information between sound events. Conversely, when β is too large, the node's information is overshadowed, leading to an excessive smoothing phenomenon.

The experimental findings reveal that even when β is set to a large value, the model's performance does not deteriorate significantly. This observation suggests that the graph structure created in this study, which employs the calculated conditional probability value as the edge weight, effectively captures diverse information by assigning varying weights to each edge during the aggregation of neighboring node information. In comparison to the binary correlation matrix utilized in previous works [14] and [17], the adoption of the calculated conditional probability value as the edge weight in the graph structure partly mitigates the issue of over-smoothing.

Experimental effect of sound event window size on model performance

In this experiment, we conduct comparative experiments on two different datasets to evaluate the performance of the G-CRNN (CS) model under three different sizes of sound event

Table 4 Model detection ability for different sound event window sizes in the both datasets

Dataset	Window size	F_1 score (%) ↑	ER metric (%) ↓	Precision (%) ↑	Recall (%) ↑
Residential area	–	48.48 ± 3.73	84.40 ± 0.92	**53.93** ± 3.89	45.37 ± 8.84
	30 s	50.74 ± 4.54	84.44 ± 3.38	51.18 ± 0.95	51.25 ± 9.40
	60 s	**51.35** ± **2.42**	**83.65** ± **0.71**	51.41 ± **0.69**	**51.61** ± **5.36**
	90 s	49.85 ± 5.34	87.86 ± 3.22	49.97 ± 4.46	50.95 ± 10.1
Street	–	46.71 ± 1.93	82.26 ± 2.47	47.44 ± 1.75	46.19 ± 3.64
	30 s	48.07 ± 1.33	83.17 ± **1.73**	47.55 ± **1.41**	**48.77** ± 3.12
	60 s	**48.62** ± **0.74**	**79.60** ± 2.90	**50.37** ± 2.56	47.12 ± **1.23**
	90 s	46.87 ± 1.49	83.57 ± 3.53	47.96 ± 2.28	45.95 ± 2.27

↑ indicates that higher is better for this index; ↓ indicates that lower is better for this index; ± is followed by standard deviation; and – indicates G-CRNN(C) without a sound event window

windows, which are 30 s, 60 s, and 90 s, respectively. Table 4 shows the performance comparison of the models on the two datasets under different window sizes. The final detection performance of the model is determined by calculating the mean and standard deviation of the results of all cross-validation sets for both datasets, and the optimal performance is marked in bold font.

In Table 4, the proposed sound event window size has a relatively obvious impact on the model's performance. In summary, the model performs best when the window size is set to 60 s, and there is little difference in model performance when the window size is set to 30 s. However, when the window size increases to 90 s, the model's performance significantly degrades. This suggests that when the window is too large (90 s), the model may introduce too much noisy relation data, which may interfere with the model's learning of beneficial sound event relations, leading to performance degradation. On the contrary, when the window is too small (30 s), although the noise relationship data is greatly reduced, some useful sound event relationship information may be ignored, so the model is insufficient in obtaining comprehensive sound event relationship information, resulting in a slight decrease in performance. When the window size is moderate (60 s), the model can effectively balance the acquisition of beneficial sound event relationship information and the introduction of noise relationship. By setting the filtering threshold α, the model can effectively remove most of the noise relationship data while retaining beneficial sound event relationship information, thereby improving its performance. A more detailed experimental study of window sizes will be conducted in the future to determine the optimal sound event window size.

Compared with the G-CRNN(C) model using only co-occurrence relationships, the G-CRNN(CS) model with a sound event window improves performance when the window size is between 30 and 60 s, and even the worst-performing model with a window size of 90 s achieves comparable performance as G-CRNN(C). This shows that the introduction of a sound event window significantly improves the comprehensiveness of sound event relationship information obtained by the model, thus effectively improving the detection performance of the model. Especially, the recall rate is almost significantly improved after the introduction of the sound event window method, which proves that the introduction of the sound event succession occurrence relationship can effectively enhance the detection ability of the model.

Model parameters and real-time experiments

To examine the influence of the proposed multi-modal information acquisition method on model parameter count, the number of model parameters for CRNN and G-CRNN is provided. The event relation learning module (GCN) consists of 133, 120 parameters, while both the acoustic feature learning module (CRNN) and G-CRNN have 490, 182 parameters each.

The training and testing process of G-CRNN involves training the event relation learning module once in advance to obtain the event relation feature sequence, which is then saved for future use. There is no need to further train the event relation module, so the actual number of parameters in G-CRNN remains the same as in CRNN. It only involves performing a Hadamard product calculation during multimodal fusion.

To evaluate the real-time processing ability of the model and the impact of the multimodal fusion method on the detection time, the index FPS (frames per second) is used in this study. FPS can measure the processing speed of the model in real applications, which is particularly important for real-time deployment. FPS is calculated as follows:

$$FPS = \frac{frameCount}{elapsedTime} \tag{12}$$

Table 5 Real-time detection capability of different models in both datasets

Dataset	Residential area	Street
Model	FPS (f/s) ↑	FPS (f/s) ↑
CRNN	2363.16 ± 18.82	**2690.86 ± 10.21**
G-CRNN(C)	2364.16 ± 16.98	2687.67 ± 11.03
G-CRNN(CS)	**2363.29 ± 13.03**	2688.85 ± 11.42

↑ indicates that higher is better for this index; ± is followed by standard deviation

This paper conducts comparative experiments on two datasets to evaluate the real-time processing performance of CRNN, G-CRNN (C), and G-CRNN (CS), respectively. Table 5 shows the comparison of the real-time processing capabilities of different models on the two datasets. We evaluate the real-time performance of each model five times and calculate the mean and standard deviation, and the optimal performance is marked in bold font.

As shown in Table 5, the proposed sound event detection model has a high FPS and a low standard deviation, which proves that the model performs well and stably in real-time performance and can meet the real-time deployment requirements in traffic scenarios. It is worth noting that although the multimodal fusion operation is added to the model, it has little impact on the real-time performance of the model. This is because the feature sequence of sound event relations can be learned and saved in advance by the event relation learning module, and only one additional Hadamard product calculation is needed in the multimodal fusion stage, so the processing time will not be significantly increased.

Conclusions

Sound serves as a valuable source of information for perceiving the surrounding environment. By analyzing the characteristics of sound signals and the interrelation between sound events, the ability to detect sound events in complex driving environments is enhanced. This study proposes a graph neural network-based method for sound event detection in traffic scenes, aiming to capture multi-modal information. Detection performance is significantly improved by effectively integrating the multi-modal information and acoustic features of sound events. Experimental results demonstrate that introducing multi-modal information through the graph neural network leads to improved performance across all metrics. Specifically, the successive occurrence relationship information of sound events obtained through the sound event window approach further enhances the performance of the model based on co-occurrence relationships. This highlights the benefits of incorporating multi-modal information, enabling the model to leverage valuable prior knowledge and enhance its performance effectively. Additionally, the proposed sound event window method extracts both co-occurrence and successive occurrence relationships, addressing the issue of sparse graph structures caused by limited correlation information when relying solely on co-occurrence relationships. This method improves the efficiency of node information transmission and learning while providing a more comprehensive representation of relationship information, thereby further enhancing the model's performance. So, this sound event detection method can enhance the vehicle's environmental perception capability through the analysis of sound signals, thereby further advancing the level of intelligence in autonomous vehicles.

When applying the proposed sound event detection system to real traffic scenarios, it may encounter the challenges of environmental noise interference, overlapping sound events, data acquisition and annotation, and real-time requirements. These factors need further investigation and optimization to improve the robustness and real-time performance of the system. Future research work will be devoted to deepening event relationship learning by conducting more detailed experiments, analyzing the performance of different sizes of sound event windows, and adding attention mechanisms for specific rare events to improve detection accuracy. At the same time, we will optimize the graph network and research and develop more advanced graph network structures to extract and represent the relationship information between sound events more effectively.

Acknowledgements This work is supported by the Basic Scientific Research Project of Colleges and Universities of Liaoning Provincial Department of Education (LJKZ0338), the Science and Technology Innovation Support Project of Guangdong Province (STKJ2023071), and the Science and Technology Plan Project of Huludao City (2023JH(1)4/02b).

Data availability The two datasets that support the findings of this study are available at https://webpages.tuni.fi/arg/datasets.

Declarations

Conflict of interest The authors assert that there are no conflicts of interest in relation to the publication of this paper.

References

1. Chen Y, Zhang Y, Duan Z (2017) DCASE2017 sound event detection using convolutional neural network. Detection and classification of acoustic scenes and events
2. Zhou J (2017) Sound event detection in multichannel audio LSTM network. Detection and classification of acoustic scenes and events
3. Cakır E, Parascandolo G, Heittola T, Huttunen H, Virtanen T (2017) Convolutional recurrent neural networks for polyphonic sound event detection. IEEE/ACM Trans Audio Speech Lang Process 25(6):1291–1303. https://doi.org/10.1109/TASLP.2017.2690575
4. Lu R, Duan Z (2017) Bidirectional GRU for sound event detection. Detection and classification of acoustic scenes and events, pp 1–3
5. Xia W, Koishida K (2019) Sound event detection in multichannel audio using convolutional time-frequency-channel squeeze and excitation. https://doi.org/10.48550/arXiv.1908.01399
6. Watcharasupat KN, Nguyen TNT, Nguyen NK, Lee ZJ, Jones DL, Gan WS (2021) Improving polyphonic sound event detection on multichannel recordings with the Sørensen-dice coefficient loss and transfer learning. https://doi.org/10.48550/arXiv.2107.10471
7. Wang X, Zhang X, Zi Y, Xiong S (2022) A frame loss of multiple instance learning for weakly supervised sound event detection. In: ICASSP 2022–2022 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, pp 331–335. https://doi.org/10.1109/ICASSP43922.2022.9746435
8. Feroze K, Maud AR (2018) Sound event detection in real life audio using perceptual linear predictive feature with neural network. In: 2018 15th international Bhurban conference on applied sciences and technology (IBCAST). IEEE, pp 377–382. https://doi.org/10.1109/IBCAST.2018.8312252
9. Adavanne S, Virtanen T (2017) A report on sound event detection with different binaural features. https://doi.org/10.48550/arXiv.1710.02997
10. Ke Q, Jing X, Woźniak M, Xu S, Liang Y, Zheng J (2024) APGVAE: adaptive disentangled representation learning with the graph-based structure information. Inf Sci 657:119903. https://doi.org/10.1016/j.ins.2023.119903
11. Tonami N, Imoto K, Yamanishi R, Yamashita Y (2021) Joint analysis of sound events and acoustic scenes using multitask learning. IEICE Trans Inf Syst 104(2):294–301. https://doi.org/10.1587/transinf.2020EDP7036
12. Komatsu T, Watanabe S, Miyazaki K, Hayashi T (2022) Acoustic event detection with classifier chains. arXiv:2202.08470. https://doi.org/10.21437/Interspeech.2021-2218
13. Wang H, Zou Y, Chong D, Wang W (2020) Modeling label dependencies for audio tagging with graph convolutional network. IEEE Signal Process Lett 27:1560–1564. https://doi.org/10.1109/LSP.2020.3019702
14. Sun Y, Ghaffarzadegan S (2020) An ontology-aware framework for audio event classification. In: ICASSP 2020–2020 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, pp 321–325. https://doi.org/10.1109/ICASSP40776.2020.9053389
15. Imoto K, Kyochi S (2019) Sound event detection using graph Laplacian regularization based on event co-occurrence. In: ICASSP 2019–2019 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, pp 1–5. https://doi.org/10.1109/ICASSP.2019.8683708
16. Nt H, Maehara T (2019) Revisiting graph neural networks: all we have is low-pass filters. https://doi.org/10.48550/arXiv.1905.09550
17. Chen ZM, Wei XS, Wang P, Guo Y (2019) Multi-label image recognition with graph convolutional networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 5177–5186. https://doi.org/10.1109/CVPR.2019.00532
18. Luan S, Hua C, Lu Q, Zhu J, Zhao M, Zhang S, Precup D (2022) Revisiting heterophily for graph neural networks. Advances in neural information processing systems, vol 35, pp 1362–1375. https://doi.org/10.48550/arXiv.2210.07606
19. Luan S, Hua C, Xu M, Lu Q, Zhu J, Chang X W, Precup D (2024) When do graph neural networks help with node classification? Investigating the homophily principle on node distinguishability. Advances in Neural Information Processing Systems, vol 36. https://doi.org/10.48550/arXiv.2304.14274
20. Luan S, Zhao M, Hua C, Chang X W, Precup D (2020) Complete the missing half: augmenting aggregation filtering with diversification for graph convolutional networks. https://doi.org/10.48550/arXiv.2008.08844
21. Dong W, Wu J, Zhang X, Bai Z, Wang P, Woźniak M (2022) Improving performance and efficiency of graph neural networks by injective aggregation. Knowl Based Syst 254:109616. https://doi.org/10.1016/j.knosys.2022.109616
22. Mesaros A, Heittola T, Virtanen T (2016) TUT database for acoustic scene classification and sound event detection. In: 2016 24th European signal processing conference (EUSIPCO). IEEE, pp 1128–1132. https://doi.org/10.1109/EUSIPCO.2016.7760424
23. Mesaros A, Heittola T, Diment A, Elizalde B, Shah A, Vincent, E, Virtanen T (2017) DCASE 2017 challenge setup: tasks, datasets and baseline system. In: DCASE 2017-workshop on detection and classification of acoustic scenes and events. http://urn.fi/URN:ISBN:978-952-15-4042-4
24. Mesaros A, Heittola T, Virtanen T (2016) Metrics for polyphonic sound event detection. Appl Sci 6(6):162. https://doi.org/10.3390/app6060162
25. Venkatesh S, Moffat D, Miranda ER (2022) You only hear once: a yolo-like algorithm for audio segmentation and sound event detection. Appl Sci 12(7):3293. https://doi.org/10.3390/app12073293