symmetry

MDPI

Article

Local Time-Frequency Feature Fusion Using Cross-Attention for Acoustic Scene Classification

Rong Huang [1,*], Yue Xie [2] and Pengxu Jiang [3]

[1] Information Construction and Management Office, Nanjing University of Posts and Telecommunications, Nanjing 210049, China
[2] School of Communication and Artificial Intelligence, School of Integrated Circuits, Nanjing Institute of Technology, Nanjing 211167, China; xieyue0109@njit.edu.cn
[3] School of Information Science and Engineering, Southeast University, Nanjing 210018, China; px20115c@163.com
* Correspondence: huangrong@njupt.edu.cn

Abstract: To address the interdependence of local time-frequency information in audio scene recognition, a segment-based time-frequency feature fusion method based on cross-attention is proposed. Since audio scene recognition is highly sensitive to individual sound events within a scene, the input features are segmented into multiple segments along the time dimension to obtain local features, allowing the subsequent attention mechanism to focus on the time slices of key sound events. Furthermore, to leverage the advantages of both convolutional neural networks (CNNs) and recurrent neural networks (RNNs), which are mainstream structures in audio scene recognition tasks, this paper employs a symmetry structure to separately obtain the time-frequency features output by CNNs and RNNs and then fuses the two sets of features using cross-attention. Experiments on the TUT2018, TAU2019, and TAU2020 datasets demonstrate that the performance of this algorithm improves the official baseline results by 17.78%, 15.95%, and 20.13%, respectively.

Keywords: audio scene recognition; feature fusion; cross-attention; convolutional neural networks; recurrent neural networks

check for updates

Academic Editor: Zhengqiu Zhang

Received: 25 October 2024
Revised: 23 December 2024
Accepted: 28 December 2024
Published: 30 December 2024

Citation: Huang, R.; Xie, Y.; Jiang, P. Local Time-Frequency Feature Fusion Using Cross-Attention for Acoustic Scene Classification. *Symmetry* **2025**, *17*, 49. https://doi.org/10.3390/sym17010049

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Acoustic scene classification (ASC) and sound event detection are important techniques for the computational analysis of natural acoustic scenes [1]. These tasks typically serve as the front end of audio processing and include the recognition of indoor scenes, outdoor scenes, public spaces, and office environments. ASC also has many application scenarios in reality. For example, in the field of medical rehabilitation, it can help cochlear implants and hearing aids better adapt to different sound scenes as well as improve the speech understanding ability and listening comfort of people with hearing impairments in various environments. In intelligent driving assistance, sound scene recognition can help vehicles better perceive the surrounding environment. For example, it can recognize the noise of road construction and the sirens of special vehicles such as ambulances to provide timely reminders and warnings to drivers. On smart wearable devices, through sound scene recognition, settings such as the ringtone, notification volume, and screen brightness of the mobile phone can be automatically adjusted. For example, the notification volume is increased on a noisy street, and it is automatically muted in a quiet conference room.

Early research on audio scene classification typically focused on the perceptual features of the human auditory system, combined with classic machine learning algorithms such as the hidden Markov model (HMM) and Gaussian mixture model (GMM). For example,

Clarkson et al. [2] calculated Mel-scale filter bank coefficients for ASC. Couvreur et al. [3] used linear predictive cepstral coding (LPCC) features and discrete HMMs to identify five types of sound events. Ahmad et al. [4] explored various 2D feature representations including the spectrogram, MFCC spectrogram, log Mel-spectrogram, and the perceptual weighted log Mel-spectrogram (PW-LMSP) for acoustic scene classification. Eronen et al. [5] proposed an ASC system based on Mel-frequency cepstral coefficients (MFCC), classifying different acoustic scenes using GMM/HMM and, in subsequent work, further classified 18 different acoustic scenes based on a richer set of acoustic features, achieving an overall accuracy of 58%. In addition to acoustic features, Heittola et al. [6] also performed scene classification using event histograms.

With the development of deep learning methods, the performance of ASC systems has significantly improved. For example, Valenti et al. [7] converted audio files into log-Mel spectrograms as input to train CNN models. Xu et al. [8] proposed an improved deep neural network (DNN) classification method, achieving performance improvements of 10.8% and 22.9% compared to classical DNN structures and GMMs. Additionally, extensions to the CNN architecture have been shown to enhance feature learning performance. For instance, Basbug et al. [9] employed a spatial pyramid pooling strategy to pool and combined feature maps at different spatial resolutions, achieving a classification accuracy of 59.5% on the DCASE dataset. Zhang et al. [10] proposed an end-to-end CNN for ASC systems. Cai et al. [11] highlighted the benefits of using separate kernels of CNNs as a more powerful and efficient design approach to ASC tasks. Temporal networks can also be applied in ASC research. For example, Zöhrer et al. [12] used gated recirculation unit (GRU) and linear discriminant analysis to classify different audio scenes. Li et al. [13] utilized a bidirectional long short-term memory network (Bi-LSTM) as a classifier to map MFCC features. Vij et al. [14] employed LSTM to learn log-Mel features.

The above studies on algorithms indicate that convolutional neural networks (CNNs) and recurrent neural networks (RNNs) are beneficial for enhancing the performance of ASC. However, these algorithms are typically used independently as either CNNs or RNNs. To combine the advantages of both, some studies have connected CNNs and RNNs for ASC tasks [15]. In contrast, this paper adopts a parallel approach, first extracting the embedded features from both CNN and RNN separately. Then, it utilizes cross-attention to fuse the embedded features, using the temporal features output by the RNN as the query to provide a time-dimension weighting for the time-frequency features extracted by the CNN. Finally, a classifier built with a fully connected neural network is used to obtain audio scene categories. On the TUT2018 dataset, this algorithm outperforms the concatenated RNN–CNN algorithm [15] by 3.88%.

2. Network Structure

The proposed symmetrical CNN–RNN model based on cross-attention is illustrated in the Figure 1. The model uses log-Mel features as input; however, directly using these features may struggle to capture the correlations of different sound events specific to certain scenes, which can limit the performance of the ASC system. To address this, this study employs time-domain segmented Log-Mel features as inputs for both CNN and RNN, denoted as f_{CNN} and ϕ_{rnn}, respectively, the input segment-level features denoted as $X = \{x_1, x_2, \ldots, x_n\}$. The two modules help the ASC system obtain high-level segment features based on time-frequency and temporal information, which can be represented as

$$f_{cnn}(X) = \{f_{cnn}(x_1), f_{cnn}(x_2), \ldots, f_{cnn}(x_N)\} \quad (1)$$

$$\phi_{rnn}(X) = \{\phi_{rnn}(x_1), \phi_{rnn}(x_2), \ldots, \phi_{rnn}(x_N)\} \quad (2)$$

where N is the number of segments in the time domain of the spectrogram, and C is the dimension of the outputs from different modules, meaning that the number of hidden units in the RNN is set to be consistent with the output dimension of the CNN. The CNN is composed of residual convolutional blocks, while the RNN consists of 2 layers of GRU. Their outputs $f_{cnn}(X)$ and $\phi_{rnn}(X)$ have symmetry in the embedded space. They have not only the same dimension but also complementary symmetry at the information level. Then, the features are fused through cross-attention, where the features from the RNN serve as the query in the attention mechanism. Based on the attention scores, the fused features are weighted and summed along the time dimension. Finally, the classification results for ASC are obtained through a classifier constructed with fully connected neural networks.

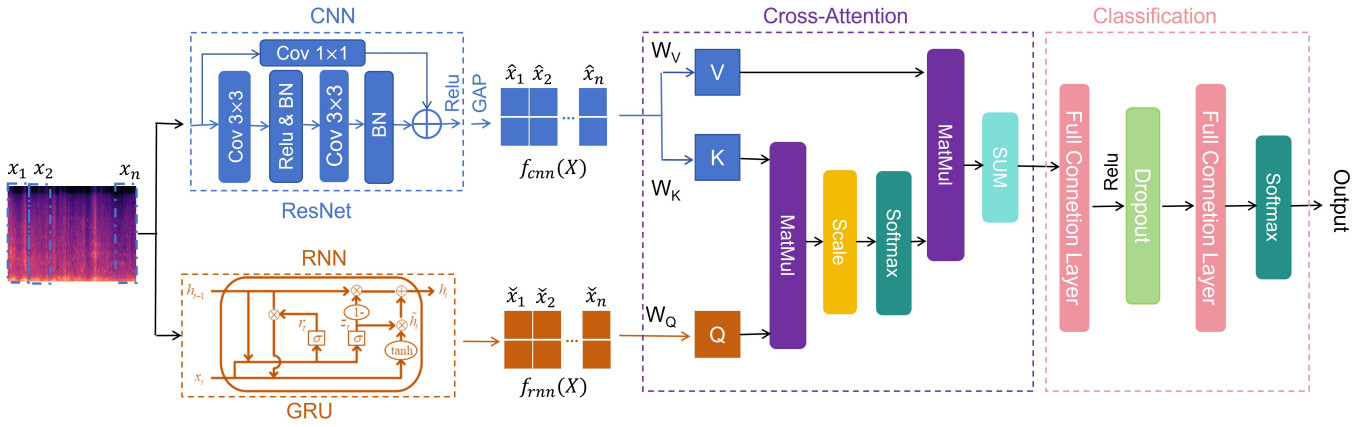


Figure 1. Network structure.

3. Algorithm

3.1. Segment-Level Features

Log-Mel features are employed as the input of the model, expressing the frequency distribution of sound signals in the cepstral domain. Under normal circumstances, the frequency-domain signal obtained based on STFT contains a lot of redundant information, and the Mel filter bank is needed to simplify the amplitude in the frequency domain. This is mainly achieved by simulating the human ear's ability to distinguish high and low frequencies of sound. In the Mel frequency scale, the resolution of lower frequencies is higher, while the resolution of higher frequencies is lower, which makes the Mel frequency scale more in line with the characteristics of human hearing. The Mel filter is a set of triangular filters described in Equation (3). These filters are used to divide the original audio signal into different frequency bands and extract the energy of each frequency band.

$$H_m(k) = \begin{cases} 0, k < f(m-1) \\ \frac{k-f(m-1)}{f(m)-f(m-1)}, f(m-1) \le k \le f(m) \\ \frac{f(m+1)-k}{f(m+1)-f(m)}, f(m) < k \le f(m+1) \\ 0, k > f(m+1) \end{cases} \tag{3}$$

The Mel filter bank is applied to the energy spectrum to obtain Mel features.

$$\mathrm{Y}_t(m) = \sum_{k=1}^{N} H_m(k)|X_t(k)|^2 \tag{4}$$

where $X_t(k)$ is the result of STFT. In addition, log-Mel features are obtained through the logarithm of the Mel spectrum. This helps compress high-amplitude values and increase the sensitivity to low-amplitude values, which is more in line with human auditory perception. Since the human ear's perception of sound is not linear, and Mel features based

on logarithms can better describe the nonlinear relationship between frequency points, log-Mel features are often used as input features in ASC tasks based on deep learning.

The ASC system typically needs to consider the large number of complex local environments in acoustic data. However, a segment of an audio scene may contain multiple sound events, and audio recorded in different scenes may have similar sound events. For example, the rumble of a tram may occur both inside the carriage and at the station, music may be present in both a cafe and a restaurant, and birdsong may be heard in both a park and on the street. In most cases, audio scenes usually include sound events, noise, or echoes, as well as their combinations. There may even be audio segments that do not contain any obvious events, such as long periods of silence or quiet. Generally, people can perceive the scene of a park through the sounds of birds and streams, but if there is no event information in the park, the ASC system struggles to distinguish a park from another outdoor scene. Therefore, audio segments may contain specific sounds that represent a scene or may only include common sounds that can occur in multiple scenes. If the audio contains sounds specific to a certain scene, the scene can be identified more accurately within the same category; however, if not, it becomes difficult to make distinctions among these sounds.

In a long-duration recorded audio dataset, there may be multiple sound events. As shown in Figure 2, each event information is sufficient to express multiple corresponding environments, and there can be common features between different scenes. For example, both parks and squares may contain a lot of quiet segments, and the rumble of cars can be heard in both traffic streets and squares. Therefore, simply identifying a single event may make it difficult to accurately determine the environment. Additionally, in some real recorded audio scenes, relevant information may have a short duration, such as the honking of cars at a bus station or the rumble of planes taking off at an airport. Thus, the scene information contained in a segment of audio may be limited and dispersed, meaning that there may be a lot of irrelevant information in the recorded audio. Consequently, audio data based on combinations of multiple acoustic events make it challenging for existing ASC systems to capture specific scene information, especially for temporal models, where long periods of nonsemantic environmental information may hinder their ability to effectively focus on specific event information. In light of this, to assist ASC systems in acquiring scene information, the input of the model designed in this paper consists of segment-level spectrogram features segmented along the time axis. Here, the original spectrogram can be represented as $X \in R^{T \times F}$, and the temporally correlated segment-level features can be represented as $X = \{x_1, x_2, \ldots, x_n\}$, $x_n \in R^{\frac{T}{N} \times F}$, where T and F represent the time scale and frequency scale of the spectrogram, respectively. In the experimental phase, further research was conducted on the impact of the number of segments N on system performance.

3.2. Residual Convolution

The residual neural network (ResNet) was proposed by Kaiming He and others in 2015 [16]. Its main contribution lies in identifying and addressing the "degradation phenomenon" that exists in deep neural networks. The degradation phenomenon refers to the issue where, as the depth of the network increases, the training error also increases, leading to a decline in performance. To solve this problem, He et al. introduced the concept of "shortcut connections", which allows for skip connections in a network, enabling information to be transmitted more directly between layers. This structure greatly alleviates the training difficulties experienced with deep neural networks, allowing for the construction of deeper networks while maintaining good performance. The success of ResNet laid an important foundation for subsequent research in deep learning.

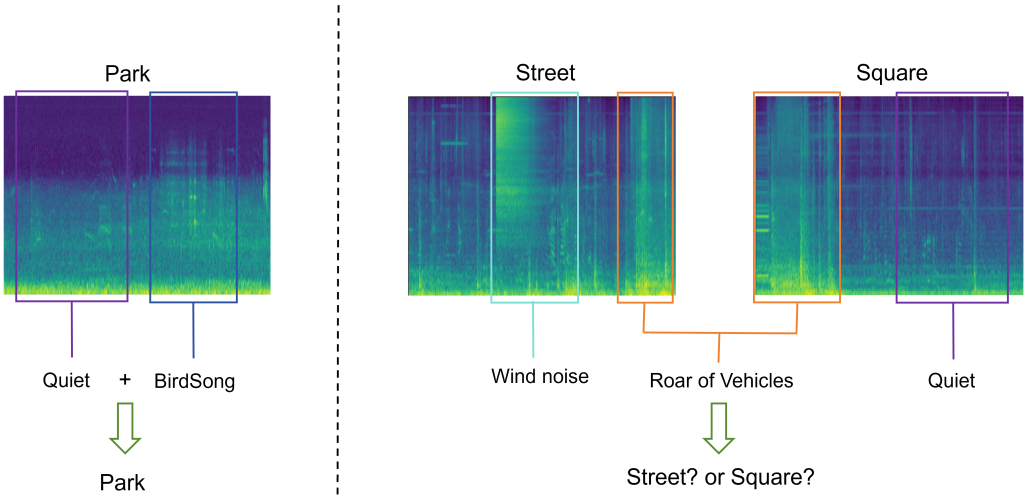


Figure 2. Multiple events included in different time domains for spectrograms.

The residual structure used in this paper is shown in the blue dashed box in Figure 1. The residual block consists of two 3×3 convolutional layers with the same number of output channels, each followed by a batch normalization layer and a ReLU activation function. Through a cross-layer data pathway, the input is added directly before the final ReLU activation function, skipping the two convolution operations within the residual block. This design requires the outputs of the two convolutional layers to have the same shape as the input, ensuring that the output of the second convolutional layer matches the original input shape for addition. When the number of channels differs, an additional 1×1 convolutional layer is needed to transform the input into the required shape before performing the addition. The principle is that the 1×1 convolutional layer does not alter anything in the spatial dimension, and it primarily changes the channel dimension.

3.3. Gated Recurrent Unit

The gated recurrent unit (GRU) proposed by Cho was originally applied in the field of machine translation [17]. Its core idea is that feature vectors at different distances in the hidden layer have varying impacts on the current hidden state, with the influence diminishing as the distance increases. The GRU introduces a gating mechanism to control the flow of information, effectively capturing long-term dependencies. Its main components include the reset gate and the update gate, which allow the model to flexibly choose which information to retain and which to update, resulting in excellent performance when handling sequential data. The specific computation methods can be found in Equations (5)–(8), and a structural diagram of the GRU is illustrated in the brown-colored dashed box in Figure 1.

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t]) \tag{5}$$

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t]) \tag{6}$$

$$\tilde{h}_t = \tanh(W_h \cdot [r_t \odot h_{t-1}, x_t]) \tag{7}$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t \tag{8}$$

where $\odot$ represents the Hadamard product. W_z, W_r, and W_h are weight matrices that need to be trained. z_t is the update gate, which controls the amount of historical information retained at the current time step, thereby helping the RNN remember long-term dependencies. r_t is the reset gate. When its value is 0, it indicates that it is turned off. In this case, the candidate hidden layer output is determined solely by the current input and is inde-

pendent of historical outputs. This allows the hidden state to effectively discard irrelevant information from the historical data, resulting in a more robust compressed representation.

3.4. Cross-Attention

The implementation of cross-attention is derived from the self-attention mechanism [18], but it considers data from different sources when processing inputs. In the self-attention mechanism, the query, key, and value typically come from the same data source. In this paper, the output features of the GRU are used as the query, while the output features of the CNN are used as the key and value. The calculation method is as follows:

$$Q = \phi_{rnn}(X)W_Q \tag{9}$$

$$K = f_{rnn}(X)W_K \tag{10}$$

$$V = f_{rnn}(X)W_V \tag{11}$$

$$Score = \text{softmax}(\frac{QK^H}{\sqrt{N}}) \tag{12}$$

$$Atten = \sum_T Score \times V \tag{13}$$

The Q, K, and V are obtained through projection transformations by $\phi_{rnn}(X)$ and $f_{rnn}(X)$, where $W_Q, W_K, W_V \in R^{C\times C}$ are parameters to be learned by the network and have the same dimension. In Equation (12), the attention scores are obtained using the softmax function, where N corresponds to the number of time segments. To prevent the inner product from becoming too large when calculating the the multiplication operation of matrix Q and matrix K, a scaling factor is applied. Finally, the attention mechanism scores are used to perform a weighted sum of the V values along the time dimension.

4. Experiments and Discussion

The model structure is shown in Figure 1. The detailed parameter settings are as follows: The RNN structure encompasses 2 layers of GRU modules, with 512 and 128 hidden layer units. The CNN consists of two residual block structures, which possess 64-channel and 32-channel outputs. These two residual blocks are interconnected by an average pooling layer of 2×2 size. The size of all convolution kernels is uniformly 3×3. In the classification network, the first fully connected layer comprises 128 hidden layer units, and the final fully connected layer has 10 hidden layer units corresponding to the number of classifications. The hyperparameter settings of the network include epoch of 128, batch size of 32, learning rate of 0.001, and MomentumOptimizer as the optimizer.

4.1. Datasets

To validate the effectiveness of the proposed algorithm, experiments were conducted on three databases provided by the Detection and Classification of Acoustic Scenes and Events Challenge [19].

TUT 2018 is a publicly available dataset recording ten different acoustic scenes from six European cities, specifically, airport, indoor shopping mall, subway station, pedestrian street, public square, traffic street, tram, bus, subway, and urban park. Each acoustic scene in TUT 2018 consists of 864 segments, totaling 8640 audio clips. The training set contains 6122 samples, while the test set includes 2518 samples.

TAU 2019 is an extended version of the TUT 2018 dataset, expanded from six European cities to twelve. However, the development dataset includes only ten of these cities and contains the same ten acoustic scenes as the TUT 2018 dataset. In the officially designated

training and test sets, the training set consists solely of audio data from nine cities to evaluate the system's generalization capabilities. This dataset comprises 40 h of recorded data across 14,400 audio files, with 9185 files allocated to the training set and the remaining files used as the test set.

TAU 2020 builds upon the TAU 2019 dataset, further expanding the acoustic scene classification through the use of four different devices to simultaneously record data. This includes three portable devices, such as smartphones and cameras, as well as synthetic data created from audio recorded by multiple devices. TAU 2020 contains data from ten acoustic scenes across twelve European cities, recorded at a sampling rate of 44.1 kHz, with a total duration of 64 h. The dataset includes 13,965 samples for training and 2970 samples for testing.

4.2. The Impact of Time Segmentation

To test the impact of segment-level feature model inputs at different time scales on system performance, the maximum number of segments N for the log-Mel features based on the time domain is set to 10, indicating that global features are used as input. The models tested include

- CNN+SA: An independent CNN model followed by self-attention;
- GRU+SA: An independent GRU model followed by self-attention;
- CNN-GRU+CA: A proposed parallel CNN and GRU model followed by cross-attention.

When N = 1 , the attention mechanism uses frame-level features as input, since log-Mel features require audio to be segmented into frames for feature extraction. By comparing these models, the impact of different time-scale feature inputs on system performance can be evaluated, as shown in Figures 3–5.

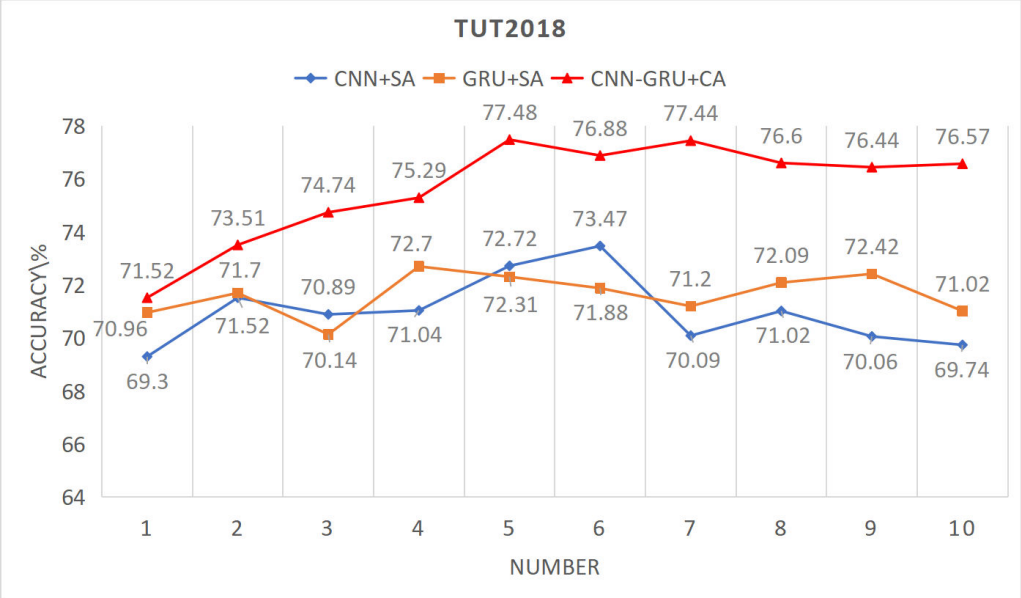


Figure 3. Time segmentation on TUT2018.

For the independent CNN, the trend observed across the three databases indicates that as the number of segments increases, the recognition performance initially improves but declines when the number of segments becomes too high. This is because each sound scene typically consists of only a few types of sound events; for example, in a park scene, there may only be bird chirping and silence as sound events. When the number of segments is excessive, the subsequent attention scores become dispersed, making it difficult to focus on the key sound events. However, the optimal number of segments is uncertain. The experiments conducted on the TUT2018, TAU2019, and TAU2020 datasets showed that the

best performance was achieved with segment counts of 6, 4, and 5, yielding recognition rates of 73.47%, 75.71%, and 68.67%, respectively.

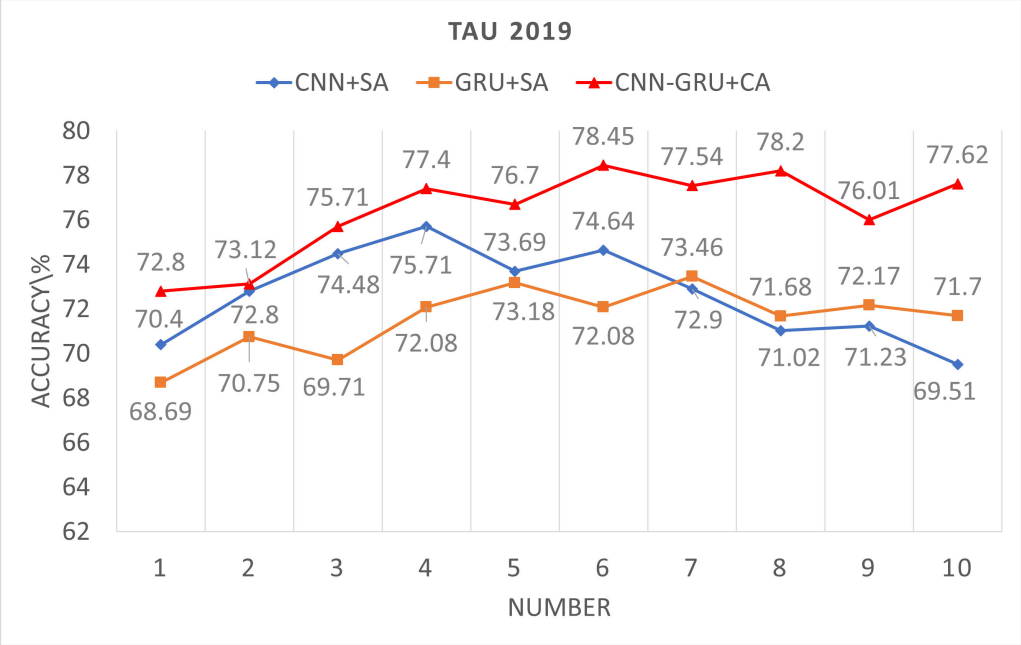


Figure 4. Time segmentation on TAU2019.

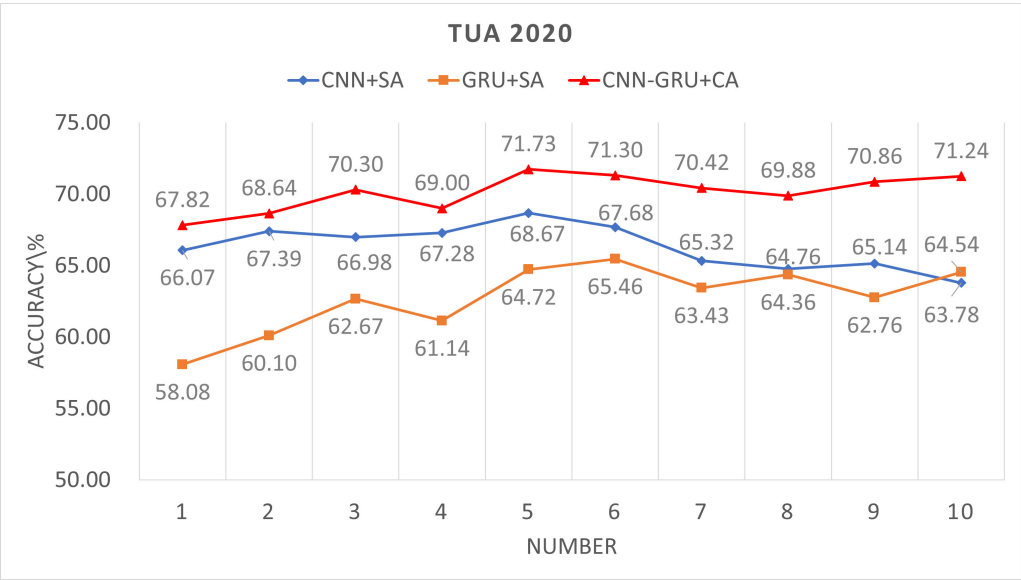


Figure 5. Time segmentation on TAU2020.

For the independent GRU network, as the number of segments increases, the performance of the GRU shows some improvement and eventually stabilizes. This means that an excessive number of segments does not lead to further performance enhancement, but it does not cause a significant decline like in CNNs. This is because GRUs have memory capabilities, giving them an advantage in processing sequential data. On the TUT2018, TAU2019, and TAU2020 datasets, the best performances achieved by the GRU were 72.7%, 73.46%, and 65.46%, respectively. Although the best performance of the GRU was not as high as that of the CNN, this is because CNNs balance both temporal- and frequency-domain computations more effectively. Therefore, in the research on ASC, there has been relatively more focus on CNNs.

The parallel CNN–GRU network, utilizing cross-attention for feature fusion, demonstrates significantly better performance compared to the independent CNN and GRU models. On the TUT2018, TAU2019, and TAU2020 datasets, the best performance achieved was 77.48%, 78.45%, and 71.73%, respectively. This approach combines the CNN's ability to process time-frequency information with the GRU's memory capability for sequential information. In the temporal dimension, the sequential features provided by the GRU are used to apply attention weighting to the time-frequency features extracted by the CNN. This enhances the characteristics of the key sound events in the audio scene, thereby improving the overall system performance.

4.3. Confusion Matrix

In order to describe the recognition results among various categories, we conducted analyses through the confusion matrix, as shown in Tables 1–3.

It can be seen from all three tables that misidentifications were prone to occurring between the categories of "Subway" and "Tram". For example, the probability of "Tram" being misidentified as "Subway" reached 10.71%, 11.69%, and 15.82% on the TUT2018, TAU2019, and TAU2020 datasets, respectively, because both of these scenarios belong to scenes inside transportation carriages and are extremely similar. Therefore, they were also likely to be confused with the "Bus" scene. When the differences between scenes were relatively large, there was a relatively high recognition accuracy. For instance, "Tram" was never misjudged as "Pedestrian Street" for the three datasets. However, the recognition rate of "Pedestrian Street" was not high on the three datasets, with only 52.3% on TAU2019 and 48.48% on TAU2020. The sounds in "Pedestrian Street" are mainly those of people talking and communicating, but human voices also frequently appear in other scenes, such as in the "Public Square" and "Indoor Shopping Mall" scenes. Therefore, "Pedestrian Street" could be misjudged as other categories.

Table 1. Confusion matrix on TUT2018 (%).

Label \ Predict	Airport	Bus	Subway	Subway Station	Urban Park	Public Square	Indoor Shopping Mall	Pedestrian Street	Traffic Street	Tram
Airport	82.55	0	0	12.3	0	0	1.98	2.38	0.79	0
Bus	0	63.02	12.3	0.4	0.79	0	0	0	0	23.49
Subway	0	0.79	80.16	7.54	0	0	0	0.4	0.4	10.71
Subway Station	3.17	0	3.97	90.87	0	0	0.4	1.19	0	0.4
Urban Park	0.4	0	0	1.99	83.67	6.37	0.8	4.38	1.99	0.4
Public Square	2.38	0	0	1.98	6.75	60.37	0	13.89	14.23	0.4
Indoor Shopping Mall	22.72	0	0	1	0	1.19	69.14	5.95	0	0
Pedestrian Street	7.57	0	0	0.79	0	12.35	0.79	75.32	3.17	0
Traffic Street	0	0	0	0.4	0.79	2.78	0	5.56	90.47	0
Tram	0	2.39	16.73	0.4	0.4	0	0	0	0.8	79.28

Table 2. Confusion matrix on TAU2019 (%).

Label \ Predict	Airport	Bus	Subway	Subway Station	Urban Park	Public Square	Indoor Shopping Mall	Pedestrian Street	Traffic Street	Tram
Airport	79.65	0.19	0.19	8.25	0.77	0.58	9.02	0.77	0.58	0
Bus	0	88.34	2.3	0.19	0.57	0	0	0	0.19	8.41
Subway	0	5.94	77.01	4.21	0	0.96	0.19	0	0	11.69
Subway Station	1.15	0.19	6.5	85.09	0	0.38	2.3	0.38	0.57	3.44
Urban Park	0	0	0	1.34	88.88	3.07	0.77	0.38	4.6	0.96
Public Square	1.34	0.77	0.19	1.34	6.72	71.69	0.67	7.68	7.87	1.73
Indoor Shopping Mall	15.08	0	0.04	5.36	0	4.2	66.6	7.42	0.19	1.11
Pedestrian Street	4.4	0	0	1.34	1.34	36.4	2.11	52.3	2.11	0
Traffic Street	0	0.77	0	0.77	0.77	6.9	0	0.96	89.25	0.58
Tram	0	5.36	7.85	0.96	0.19	0	0	0	0	85.64

Table 3. Confusion matrix on TAU2020 (%).

Label \ Predict	Airport	Bus	Subway	Subway Station	Urban Park	Public Square	Indoor Shopping Mall	Pedestrian Street	Traffic Street	Tram
Airport	63.1	0.34	0.67	7.07	0.34	1.35	17.52	9.27	0	0.34
Bus	0	85.52	5.05	0.34	0.67	0	0	0	0	8.42
Subway	0	3.7	71.38	7.41	0	0.67	0.34	0.67	0	15.82
Subway Station	3.37	0.67	7.07	70.37	0.34	0.34	11.11	2.36	1.01	3.37
Urban Park	0.67	0.67	2.02	0	87.88	2.02	1.35	1.01	4.04	0.34
Public Square	0	1.01	0.05	3.7	12.12	58.89	1.01	9.76	12.79	0.67
Indoor Shopping Mall	15.15	0.34	0.34	5.39	0.34	0.34	69.02	9.09	0	0
Pedestrian Street	7.74	0.67	0.34	3.37	1.35	16.5	14.14	48.48	7.07	0.34
Traffic Street	0.34	0.34	0	1.01	2.36	7.41	3.03	1.01	84.51	0
Tram	0.34	7.41	9.09	2.36	0.34	1.35	0	0.67	0.34	78.11

4.4. Feature Fusion

To verify the effectiveness of feature fusion based on the cross-attention algorithm, this study compared it with the following models: The outputs of the parallel CNN-GRU structure were concatenated along the temporal dimension, and then self-attention was applied to the concatenated features for fusion, referred to as CNN-GRU+CO+SA. In terms of model structure, the CNN and GRU models were concatenated, and self-attention was applied based on the order of concatenation, referred to as CNN+GRU+SA and GRU+CNN+SA, respectively. The results are shown in Table 4.

Compared to the concatenated models, the two parallel CNN–GRU structures demonstrated better performance because they could better leverage the advantages of both the CNN and RNN. In contrast, the concatenated structure increased the depth of the network, making training more challenging and potentially causing the advantages of the preceding CNN/GRU to be lost in the subsequent GRU/CNN structure. Additionally, while CNN-GRU+CO+SA, which directly concatenated the outputs of CNN and GRU, achieved better performance than the serial structure, the differences in feature spaces between the CNN and GRU outputs reduced the expressive power of the fused features, resulting in performance that was not as good as that of CNN-GRU+CA.

Table 4. The results of the different models.

Model	TUT2018	TAU2019	TAU2020
DCASE [19]	59.7	62.5	51.6
BiLSTM-CNN [15]	73.6	-	-
Visualfy [20]	-	76.8	-
MCTA-CNN [21]	72.4	75.71	-
LCNN [22]	-	-	70.4
VilEnsemb3 [23]	-	-	70.3
GA [24]	76.7	78.2	71.3
CNN+GRU+SA	74.86	76.83	68.75
GRU+CNN+SA	73.92	76.26	68.06
CNN-GRU+CO+SA	75.34	77.38	69.86
CNN-GRU+CA	77.48	78.45	71.73

In addition, we compared our results with those in the literature. DCASE refers to the baseline results provided by the Detection and Classification of Acoustic Scenes and Events Challenge. All these studies utilized CNN structural units. On the same dataset, the performance of our proposed algorithm surpassed that of the aforementioned methods. On the TUT2018, TAU2019, and TAU2020 datasets, our method improved the official baseline results by 17.78%, 15.95%, and 20.13%, respectively.

5. Conclusions

In the task of ASC, both CNN and RNN structures have their advantages. To combine the strengths of both, we proposed a method that uses cross-attention to fuse the output features of parallel CNN–RNN structures, allowing them to form complementary information. In this paper, the output features of the GRU were used as the query in the attention mechanism to calculate attention scores for the time-frequency features from the CNN output, applying weighting along the temporal dimension. Since sound scene recognition is often highly correlated with specific sound events within the scene, we sliced the input features along the temporal dimension, enabling the subsequent attention mechanism to focus the weighting scores on these representative sound event slices. The experimental results indicated that the proposed CNN-GRU+CA model produced improved performance compared to existing algorithms. In future work, we plan to explore the impact of environmental noise and other acoustic variations on the model's performance to enhance its robustness and applicability in diverse settings.

Author Contributions: Conceptualization, R.H. and Y.X.; methodology, R.H.; validation, P.J. and Y.X.; formal analysis, R.H. and Y.X.; investigation, P.J.; writing—original draft preparation, R.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The TUT2018, TAU2019, and TAU2020 datasets are provided by the Detection and Classification of Acoustic Scenes and Events Challenge.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Surendiran, J.; Prabhakar, P.B.E.; Ibrahim, M.M.; Saritha, G.; K, S.; Vijayan, V.B. A Systemic Review on Automatic Acoustic Scene Classification. In Proceedings of the 2024 International Conference on Power, Energy, Control and Transmission Systems (ICPECTS), Chennai, India, 8–9 October 2024; pp. 1–6.
2. Clarkson, B.; Sawhney, N.; Pentland, A. Auditory context awareness via wearable computing. *Energy* **1998**, *400*, 20.
3. Couvreur, C.; Fontaine, V.; Gaunard, P.; Mubikangiey, C.G. Automatic classification of environmental noise events by hidden Markov models. *Appl. Acoust.* **1998**, *54*, 187–206. [CrossRef]
4. Abuirbaiha, R.A.A.; Lee, C.-H.; Lien, C.C. Acoustic Scene Classification Using Perceptually Weighted Log Mel Spectrogram and Buttom-Up Broadcast Neural Network. In Proceedings of the 2024 International Conference on Consumer Electronics-Taiwan (ICCE-Taiwan), Taichung, Taiwan, 9–11 July 2024; pp. 643–644.
5. Eronen, A.; Tuomi, J.; Klapuri, A.; Fagerlund, S.; Sorsa, T.; Lorho, G.; Huopaniemi, J. Audio-based context awareness-acoustic modeling and perceptual evaluation. In Proceedings of the 2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03), Hong Kong, China, 6–10 April 2003; p. V-529. [CrossRef]
6. Heittola, T.; Mesaros, A.; Eronen, A.J.; Virtanen, T. Audio context recognition using audio event histogram. In Proceedings of the 2010 18th European Signal Processing Conference, Aalborg, Denmark, 23–27 August 2010.
7. Valenti, M.; Diment, A.; Parascandolo, G.; Squartini, S.; Virtanen, T. DCASE 2016 acoustic scene classification using convolutional neural network. In Proceedings of the Detection and Classification of Acoustic Scenes and Events 2016, Budapest, Hungary, 3 September 2016.
8. Xu, Y.; Huang, Q.; Wang, W.; Plumbley, M.D. Hierarchical learning for DNN-based acoustic scene classification. In Proceedings of the Detection and Classification of Acoustic Scenes and Events 2016, Budapest, Hungary, 3 September 2016.
9. Basbug, A.M.; Sert, M. Acoustic Scene Classification Using Spatial Pyramid Pooling with Convolutional Neural Networks. In Proceedings of the 2019 IEEE 13th International Conference on Semantic Computing (ICSC), Newport Beach, CA, USA, 30 January–1 February 2019; pp. 128–131.
10. Zhang, L.; Han, J.; Shi, Z. Learning Temporal Relations from Semantic Neighbors for Acoustic Scene Classification. *IEEE Signal Process. Lett.* **2020**, *27*, 950–954. [CrossRef]
11. Cai, Y.; Zhang, P.; Li, S. TF-SepNet: An Efficient 1D Kernel Design in Cnns for Low-Complexity Acoustic Scene Classification. In Proceedings of the ICASSP 2024—2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Seoul, Republic of Korea, 14–19 April 2024; pp. 821–825.

12. Zöhrer, M.; Pernkopf, F. Gated recurrent networks applied to acoustic scene classification and acoustic event detection. In Proceedings of the Detection and Classification of Acoustic Scenes and Events 2016, Budapest, Hungary, 3 September 2016.
13. Li, Y.; Li, X.; Zhang, Y.; Wang, W.; Liu, M.; Feng, X. Acoustic Scene Classification Using Deep Audio Feature and BLSTM Network. In Proceedings of the 2018 International Conference on Audio, Language and Image Processing (ICALIP), Shanghai, China, 16–17 July 2018; pp. 371–374.
14. Vij, D.; Aggarwal, N. Performance evaluation of deep learning architectures for acoustic scene classification. In Proceedings of the Detection and Classification of Acoustic Scenes and Events 2017, Munich, Germany, 16 November 2017.
15. Hao, W.; Zhao, L.; Zhang, Q.; Zhao, H.; Wang, J. DCASE 2018 task 1a: Acoustic scene classification by bi-LSTM-CNN-net multichannel fusion. In Proceedings of the Detection and Classification of Acoustic Scenes and Events 2018, Surrey, UK, 19–20 November 2018.
16. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
17. Cho, K.; Van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. *arXiv* **2014**, arXiv:1406.1078.
18. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17), Long Beach, CA, USA, 4–9 December 2017; Curran Associates Inc.: Red Hook, NY, USA, 2017; pp. 6000–6010.
19. Stowell, D.; Giannoulis, D.; Benetos, E.; Lagrange, M.; Plumbley, M.D. Detection and classification of acoustic scenes and events. *IEEE Trans. Multimed.* **2015**, *17*, 1733–1746. [CrossRef]
20. Naranjo-Alcazar, J.; Perez-Castanos, S.; Zuccarello, P.; Cobos, M. DCASE 2019: CNN Depth Analysis with Different Channel Inputs for Acoustic Scene Classification. DCASE2019 Challenge, Tech. Rep. June 2019. Available online: https://dcase.community/documents/challenge2019/technical_reports/DCASE2019_Naranjo-Alcazar_13.pdf (accessed on 24 October 2024).
21. Wang, Y.; Feng, C.; Anderson, D.V. A Multi-Channel Temporal Attention Convolutional Neural Network Model for Environmental Sound Classification. In Proceedings of the ICASSP 2021—2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Toronto, ON, Canada, 6–11 June 2021; pp. 930–934.
22. Shim, H.J.; Jung, J.W.; Kim, J.H.; Yu, H.J. Capturing scattered discriminative information using a deep architecture in acoustic scene classification. *arXiv* **2020**, arXiv:2007.0463.
23. Vilouras, K. Acoustic Scene Classification Using Fully Convolutional Neural Networks and Per-Channel Energy Normalization. DCASE 2020 Challenge, Tech. Rep. June 2020. Available online: https://dcase.community/documents/challenge2020/technical_reports/DCASE2020_Vilouras_3.pdf (accessed on 24 October 2024).
24. Hasan, N.W.; Saudi, A.S.; Khalil, M.I.; Abbas, H.M. A Genetic Algorithm Approach to Automate Architecture Design for Acoustic Scene Classification. *IEEE Trans. Evol. Comput.* **2023**, *27*, 222–236. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.