Feedback Module Based Convolution Neural Networks for Sound Event Classification

GWantae Kim1, (Graduate Student Member, IEEE), David K. Han2, (Senior Member, IEEE), and Hanseok Ko1, (Senior Member, IEEE)

1School of Electrical Engineering, Korea University, Seoul 136-713, South Korea
2Department of Electrical and Computer Engineering, Drexel University, Philadelphia, PA 19104, USA

Corresponding author: Hanseok Ko (hsko@korea.ac.kr)

This work was supported by the Korea Environment Industry & Technology Institute (KEITI) through the Exotic Invasive Species Management Program funded by the Korea Ministry of Environment (MOE) under Grant 2021002280004.

ABSTRACT Sound event classification is starting to receive a lot of attention over the recent years in the field of audio processing because of open datasets, which are recorded in various conditions, and the introduction of challenges. To use the sound event classification model in the wild, it is needed to be independent of recording conditions. Therefore, a more generalized model, that can be trained and tested in various recording conditions, must be researched. This paper presents a deep neural network with a dual-path frequency residual network and feedback modules for sound event classification. Most deep neural network based approaches for sound event classification use feed-forward models and train with a single classification result. Although these methods are simple to implement and deliver reasonable results, the integration of recurrent inference based methods has shown potential for classification and generalization performance improvements. We propose a weighted recurrent inference based model by employing cascading feedback modules for sound event classification. In our experiments, it is shown that the proposed method outperforms traditional approaches in indoor and outdoor conditions by 1.94% and 3.26%, respectively.

INDEX TERMS Dual-path residual network, feedback module, recurrent inference, sound event classification.

I. INTRODUCTION

While vision provides a wealth of information about a scene, much of scene understanding can also be achieved by sounds. As such, Sound Event Classification (SEC) may play a crucial role in automatic scene understanding. Acoustic recording from a scene may reveal sound producing objects, environmental conditions such as weather, or background settings including traffic noise, or others. Thus, exploiting sound as descriptive features of a scene may enhance scene recognition or understanding. SEC also affect a wide range of applications, such as sound event detection [1], audio tagging [2], audio monitoring in smart cities [3], life assistance and healthcare [4], etc.

The goal of SEC is to recognize audio recordings of various sound-producing events which may occur indoors or outdoors. In indoor conditions, reverberation is typically the main obstacle to successful classification. In the outdoors, on the other hand, it is challenging due to background noise, e.g. conversations, machine noise, vehicle noise, etc. There have been some successful efforts in the past for dealing with these issues, however, as the severity of these challenges grows, a novel alternative SEC model structure is needed. Especially, it is necessary to study a more generalized model that works well in both indoor and outdoor conditions.

SEC is similar to Acoustic Scene Classification (ASC) and Environment Sound Classification (ESC). The Detection and Classification of Acoustic Scenes and Events (DCASE) challenge [5] prompted many innovations and novel concepts to solve the challenges associated with ASC task [6]–[8]. The concepts proposed in the challenge are useful to design a more generalized model for the SEC. For the ESC, Piczak [9] and Salamon et al. [10] developed new ESC datasets and proposed a novel neural networks architecture. Their proposed ESC concept, however, focused only on outdoor sound classification. Since the SEC model should be designed to be effective in both indoor and outdoor environments, the evaluation of the SEC model includes both types of environments. SECL_UNSMS [11], which is a SEC dataset, is suitable to evaluate the SEC model in indoor environments. Training
and developing modern systems for the SEC task requires the use of large-scale labeled SEC datasets. The large-scale labeled SEC datasets have \{audio recording, sound event label\} paired samples. Many previous models used these datasets with single-inference supervised learning. However, there are no models with multi-inference supervised learning and inference search.

Over the last few decades, several approaches have been developed for SEC problems, such as Gaussian Mixture Model(GMM) [12], Deep Neural Network(DNN)-GMM-ergodic Hidden Markov Model(HMM) combination model [13], [14] presented a combination of visual and acoustic features for audio classification tasks. While these earlier methods have shown successes and had been utilized in various applications, their performances have been well surpassed in recent years by Convolution Neural Networks (CNN) based methods.

CNN based models [15]–[18] have shown significant performance improvements in SEC. Piczak [15] designed a CNN architecture to solve the environmental sound classification task, which is a subset of SEC task. Salamon and Bello [16] improved the classification performance by changing the model size and data augmentation. Mcdonnell and Gao [17] proposed dual-path residual neural networks to solve the acoustic scene classification task, which is similar to the SEC task. Kim et al. [19] also presented a multi-band CNN architecture to solve the SEC task. Ref. [18] proposed a self-teaching method for improving generalization in sound event recognition. Recently, Transformer [20] based model [21] is proposed to classify audio signal. The model successfully applied Transformer structure for audio classification task with ImageNet [22] pre-training. Although previous studies with single-inference supervised learning have yielded good results, it is necessary to propose a more generalized model structures.

Others employed some of the latest image classification methods using 2D features [23]–[25] to improved SEC performance. While recognizing the effectiveness of the methodologies developed in computer vision, we adopted some of these techniques by deliberately selecting them based on their efficacy for applying to the SEC. As shown in their effectiveness in the computer vision field, we found recurrent inference [26] and feedback structure [27], [28] particularly effective in improving SEC performance of our neural network structure.

In cognition theory, feedback connection which link the cortical visual areas can transmit response from higher-order areas to low-order areas [29], [30]. Motivated by this phenomenon, recent studies have applied the feedback mechanism [31], [32] to network architectures. The feedback module [27], [28] is a neural network based module with a recurrent structure. The module consists of the layers of the neural network with arbitrary shape and feedback loops. In the feed-forward module, input features pass through the structure of the network only once. Unlike the feed-forward module, the input features pass through the networks several times in the feedback module. The output features of the previous feedback loop are fed into the input features of the next feedback loop. The feedback module not only has the advantage of a feedback mechanism, but also can be employed to design deeper networks without increasing network parameters. The model predicts progressively improved results as multiples of these feedback loops can be stacked effectively, thus employing a recurrent inference paradigm. Our model integrates the idea of stacked feedback loops in solving ESC to train our proposed parallel network structure in a recurrent inference fashion.

The recurrent inference is a prediction manner that predicts labels several times with recurrent structure in one training step. The recurrent inference is inspired by the recurrent inference machine [26], which is proposed to solve inverse problems. The machine is applied Recurrent Neural Network(RNN) structure, which is Turing complete [33], to establish a general framework for any kind of inverse problem. According to the results of the recurrent inference machine, it has merits in generalization performance on several tasks. In [34], they proposed the recurrent inference of time-frequency mask for monaural singing voice separation. In [27], they proposed a feedback module with recurrent inference for image super-resolution. In [35], they proposed recurrent inference layers for visual question answering. From the results of these previous studies, we hypothesize that the multi-inference supervised learning and inference search can improve generalization and robustness like a collaborative learning manner [36] compared to single-inference supervised learning. Also, the method of deriving the final answer from the values predicted by multi-inference has the same effect as the ensemble model.

Although the neural network with RNN structure has successfully conquered a lot of important tasks, it has some problems to use for recurrent inference. First, the hidden units in the RNN are limited to a fixed-length vector. This problem can lose the spatial information. Second, the shape of the RNN cell is fixed. This problem can limit the number of parameters and network shapes that can be learned. To solve these problems, we design a feedback module with weight sharing. With the feedback module, input and hidden units would not lose spatial information. Moreover, it can design arbitrary cell shapes.

In this paper, we propose a novel neural network model that aggregates dual-path residual neural networks with feedback structures to recognize indoor and outdoor sound events. We also propose a multi-inference supervised learning model and its inference searching method. The block diagram describing the proposed train/test pipeline is shown in Fig. 1. First, the features are extracted from audio data recorded in various environments. The extracted features pass through a feedback module based sound event classification model and N predictions are predicted. The model is trained by N predictions with weighted recurrent inference training and tested by N predictions with inference search. Our contributions are:
We propose a neural network structure to improve the performance of the sound event classification task recording in indoor and outdoor conditions.

We present a multi-inference supervised learning model with weighted recurrent inference-based module and an inference search module to test the model.

We evaluate the proposed training procedure using indoor and outdoor environments datasets and achieve state-of-the-art performance.

The remainder of the paper is organized as follows. The relations to prior work and the proposed method of this research are described in Section II and Section III, respectively. To evaluate our model, the experimental process and results are presented in Section IV. Conclusions are drawn in Section V.

II. FEATURE EXTRACTION

In this section, the feature extraction method is explained. In this paper, the event sound recording is converted into a Mel-Spectrogram, its delta, and delta-delta features, and they are concatenated along the channel axis. To acquire these features, we first transform the fixed-length signal in the time domain into the frequency domain using Short Time Fourier Transform (STFT) to extract the Spectrogram. Second, Spectrogram is filtered by Mel-filter bank to generate Mel-Spectrogram. Third, delta and delta-delta features are calculated from the Mel-Spectrogram.

Spectrogram is one of the most frequently used 2-D representations for time series data analysis. It contains the magnitude and phase information of local sections of the time series signal as it changes over time. The STFT of the time-series signal \( x(t) \) is

\[
STFT(\tau, f) = \int_{-\infty}^{\infty} x(t) g(\tau - t) e^{-j2\pi ft} dt
\]  

where \( \tau \) and \( f \) denote the time and frequency axis of Spectrogram, and \( g(\cdot) \) is a window function. Mel-Spectrogram can be extracted by filtering Spectrogram by the Mel-filter bank. The Mel-scale aims to mimic the non-linear human ear perception of sound, by being more discriminative at lower frequencies and less discriminative at higher frequencies. The conversion equation from frequency to Mel is:

\[
Mel = 2595 \log_{10}(1 + \frac{f}{700})
\]

For practical use, the Mel-filter bank is designed by equation 2, and applying it to Spectrogram.

The Mel-Spectrogram describes the instantaneous frequency information of the audio signal. However, audio signals are time-variant signals. Therefore, having features that can express changes over time can have good influence on classification. The delta and delta-delta features are extracting information about changes with the first difference and second difference of the Mel-Spectrogram, respectively.

After concatenating Mel-Spectrogram, its delta, and delta-delta features, the shape of the extracted feature is \([BxFxTx3]\), where \( B \) is the batch size, \( F \) is the frequency bin, and \( T \) is the time bin.

III. SOUND EVENT CLASSIFICATION MODEL

The proposed deep neural network architecture is shown in Fig. 2. The network consists of three parts: an encoder, a feedback module, and a decoder.

A. ENCODER

The encoder is based on low/high frequency path residual networks. In the encoder, the input features are separated by half along the frequency axis, thus the feature size becomes \([BxF(F/2)xTx3]\). Next, the features pass through low/high frequency residual network, which is described in Fig. 3(a), and the network outputs are concatenated along the frequency axis and passed through one convolution layer so that output feature size becomes \([BxFx(T/16)x128]\).
FIGURE 2. The proposed network architecture. In the encoder, 2D input features are separated in half along the frequency axis, pass through high/low frequency path residual networks for feature extraction, and one convolution layer for changing channel size. After that, in the feedback module, there are four feedback loops for estimate class step-by-step.

As shown in Fig. 4, the residual path of the residual block contains two pre-activation convolution layers with each layer consisting of BatchNormalization(BN)-ReLU-Convolution. The one pre-activation convolution layer is

$$I(F) = \text{Conv}(\phi(BN(F)), W)$$  \hspace{1cm} (3)

where $F$ is input features, $\phi$ is ReLU activation function and $W$ is weights of the convolution layer. This layer order has an advantage of generalization because it normalizes the very first input features and the output features are not constrained over 0 by ReLU activation function.

The BN is feature normalization method for neural networks. The mean and variance of the input features are normalized to 0 and 1, respectively. The BN is calculated by the equation

$$BN_{\gamma_B, \beta_B}(x_i) = \gamma_B \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}} + \beta_B$$  \hspace{1cm} (4)

where $x_i$ is i-th sample of the mini-batch, $\mu_B$ is mini-batch mean, $\sigma_B$ is mini-batch variance, $\epsilon$ is constant, and $\gamma_B$ and $\beta_B$ are trainable parameters. However, $\gamma$ and $\beta$ terms are not trained in the BN layers except for the first layer because of regularization effect [37].

The identity path of the residual block has an average pooling layer and a channel-padding layer. The input features are squeezed by average pooling and zero-padded to the shape of the residual path. If the stride of the residual block is $$(1, 1)$$, average pooling and channel padding layers are excepted.

$$I(F) = \begin{cases} \text{ Avg}_\text{pool}(\text{ Pad}(F)) & \text{ if } \text{ stride } = (1, 2) \\ F & \text{ if } \text{ stride } = (1, 1) \end{cases}$$  \hspace{1cm} (5)

Finally, the residual path and identity path of the residual block are summed at the end of the residual block. The output features of the residual block is

$$F_{\text{out}} = l(l(F_{\text{in}})) + I(F_{\text{in}})$$  \hspace{1cm} (6)

where $F_{\text{in}}$ is input features. The spatial and temporal features of the low/high frequency of the input data are extracted by the encoding networks.

B. FEEDBACK MODULE

We design the feedback module to train our model using a recurrent inference strategy. The feedback module consists of some neural networks layers and feedback loops. In the first loop of the feedback module, the output features of the encoder are replicated once and they are concatenated along the channel axis. After that, concatenated features are passed through the neural networks layers. The layers consists of one pre-activation convolution layer and three residual blocks, as described in Fig. 3(b). In the next loop, the output features of the previous feedback loop and the output features of the encoder are concatenated along the channel axis, and the concatenated features are passed through the neural networks layers again. In each loop, the trainable parameters of the neural networks layers are reused. Finally, the procedure is repeated by N loops, thus $N[BxFx(T/16)x128]$ features are
reset gate vector are calculated to calculate current state $h_1$ since the second dimension of the tensor represents features. The output of each loop of the feedback module and $t$ are used to calculate output states $h_1$, $h_2$, ..., $h_T$ by the one GRU model. A bidirectional GRU consists of forward GRU and backward GRU. Since the features can be extracted considering both time directions, the bidirectional GRU layers are used.

In the first and second bidirectional GRU layers, the output feature size is 256 and every GRU cell output $h_1, h_2, \ldots, h_T$ are used for the output features. In the third GRU layer, the output feature size is the number of classes and only the last GRU cell output $h_T$ is used for the output features. Finally, the output features $h_T$ are used for classification with softmax activation function.

C. DECODER

The output features of each loop of the feedback module are passed through the decoder, which is shown in Fig. 3(c). The decoder consists of a BN-convolution layer and three bidirectional Gate Recurrent Units (GRUs) [38]. In the BN-convolution layer, the channel of features is reduced to 8, and reshape the features to $[B \times T \times 8F]$. In this case, since the second dimension of the tensor represents time and the third dimension of the tensor represents features, the RNN structure can be used to extract temporal features.

The GRU is a gating mechanism in recurrent neural networks. The inputs of GRU cell in time $t$ are previous state $h_{t-1}$ and input features $x_t$. First, the update gate vector and reset gate vector are calculated to calculate current state $h_t$. The update gate vector $z_u$ is

$$z_u = \sigma(W_u x_t + U_u h_{t-1} + b_u)$$

where $W_u$, $U_u$, and $b_u$ are trainable weights, and $\sigma(\cdot)$ is sigmoid function. The reset gate vector $r_t$ is

$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r)$$

where $W_r$, $U_r$, and $b_r$ are trainable weights. Next, the candidate activation vector $\hat{h}_t$ is calculated by

$$\hat{h}_t = \tau(W_h x_t + U_h (r_t \odot h_{t-1}) + b_h)$$

where $W_h$, $U_h$, and $b_h$ are trainable weights, $\odot$ denotes Hadamard product, and $\tau(\cdot)$ is tanh function. Finally, the current state $h_t$ is

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \hat{h}_t.$$  

The current state $h_t$ is used to current output vector. The input state vectors $x_1, x_2, \ldots, x_T$ are used to calculate output state vectors $h_1, h_2, \ldots, h_T$.

D. WEIGHTED FOCAL LOSS FOR RECURRENT INFERENCE

To train the model with the $N$ output predictions, the loss functions for each feedback loop should be defined. We develop the weighted recurrent inference strategy with the order of easy to hard tasks as well as a novel loss function based on the focal loss [39] to train our model as

$$L = \sum_{i=1}^{N} -\alpha(1 - p_i)^{y_i} \log p_i,$$  

where

$$p_i = \begin{cases} p_i & \text{if } y = 1 \\ 1 - p_i & \text{otherwise} \end{cases}$$

$p_i$ is predicted values of $i$-th feedback loop and $y_i$ is ground truths. According to [39], if $y_i$ is high, the loss function only affects hard-classified samples. In contrast, if $y_i$ is low, the loss function affects both hard-classified samples and well-classified samples. Thus, by adjusting the $\gamma_1, \gamma_2, \ldots, \gamma_N$, we can define loss functions for multiple predictions individually. The final loss is summation of the losses of each feedback loop.
**E. INFEERENCE SEARCH**

In the test phase, it is necessary to determine the final prediction using the predictions from feedback loops. We present three methods depending on how you choose the output: maximum training accuracy, softmax summing, and voting. These methods are inspired by the ensemble methods. The maximum training accuracy method selects the model and feedback loop that derive the best validation accuracy. Let \( p_i^N \) is the predictions of the validation data on the i-th epoch and the n-th feedback loop, then the predictions on the i-th epoch are

\[
[ p_i^1, p_i^2, \ldots, p_i^N ] = f(x, \theta_i) \]  
\[(14)\]

where \( x \) is validation data samples, \( f(\cdot) \) is classification model, and \( \theta_i \) is model weights at the i-th epoch. Then, the optimal model weights \( \theta \) using the maximum training accuracy method are

\[
\theta = \arg\max_\theta \sum_{x=1}^X p_i^N(x) == y(x) \]  
\[(15)\]

where \( X \) is the number of validation data samples, \( y \) is ground truth label, and ’==’ operator determines whether the prediction is correct or not. If prediction is correct the value is 1, and else is 0. Note that \( i = 1 \) to \( \text{max\_epoch} \) and \( n = 1 \) to \( N \).

The softmax summing method adds the output probabilities of every feedback loop and selects the highest value. The optimal model weights \( \theta \) using the softmax summing method are

\[
\theta = \arg\max_\theta \sum_{x=1}^X \frac{\sum_{n=1}^N p_i^n(x)}{X} == y(x) \]  
\[(16)\]

The voting method counts the predictions of every feedback loop and selects the most frequent label as final predictions. The optimal model weights \( \theta \) using the voting method are

\[
\theta = \arg\max_\theta \sum_{x=1}^X \frac{\text{Vote}(p_i[x], y[x])}{X} \]  
\[(17)\]

where

\[
\text{Vote}(p_i[x], y[x]) = \begin{cases} 
1 & \text{if } p_i[x] == y[x] \\
0 & \text{else,} 
\end{cases} \]  
\[(18)\]

and \( p_i[x] \) is the most frequent label of the predictions \( p_i^1[x], p_i^2[x], \ldots, p_i^N[x] \).

**F. ALGORITHM**

The algorithms of the proposed training/test procedure are shown below. In the training phase, features are extracted and parameters are initialized. Then, training the model with the weighted focal loss. Next, the validation set is selected in the training set. The model is evaluated by the inference search and save the best model. In the test phase, features are extracted and the best model is loaded. Then the test result is predicted by the model and inference search.

**Algorithm 1 Training Procedure**

1: function TRAIN\((X, Y)\)  \(\triangleright\) X: training data, Y: ground truth
2: \(X' = \text{FeatureExtract}(X)\)
3: parameter initialization
4: for \( i = 1, 2, \ldots \) do
5: \([ p_i^1, p_i^2, \ldots, p_i^N ] = \text{Model}_0(X')\) \(\triangleright\) train
6: \(\text{loss} = \sum_{n=1}^N \text{FocalLoss}_{\theta_n}(p_i^n, Y)\)
7: Optimize \( \theta \) \(\triangleright\) \( \theta \) is model weights
8: \(X_{\text{val}}' = \text{Subset}(X')\)
9: \(Y_{\text{val}} = \text{Subset}(Y)\)
10: \([ p_i^1, p_i^2, \ldots, p_i^N ] = \text{Model}_0(X_{\text{val}}') \triangleright\) validation
11: \(\text{val. acc.} = \text{InferenceSearch}([ p_i^1, p_i^2, \ldots, p_i^N ], Y)\)
12: if \( \text{val. acc.} > \text{best acc.} \) then
13: \( \theta_{\text{best}} \leftarrow \theta \)
14: best acc. = val. acc.
15: end if
16: end for
17: Return \( \theta_{\text{best}} \)
18: end function

**Algorithm 2 Test Procedure**

1: function TEST\((X)\)  \(\triangleright\) X: test data, Y: ground truth
2: \(X' = \text{FeatureExtract}(X)\)
3: \([ p_i^1, p_i^2, \ldots, p_i^N ] = \text{Model}_{\text{best}}(X')\) \(\triangleright\) test
4: test result = InferenceSearch\(([ p_i^1, p_i^2, \ldots, p_i^N ], Y)\)
5: Return test result
6: end function

The voting method counts the predictions of every feedback loop and selects the most frequent label as final predictions. The optimal model weights \( \theta \) using the voting method are

\[
\theta = \arg\max_\theta \sum_{x=1}^X \frac{\text{Vote}(p_i[x], y[x])}{X} \]  
\[(17)\]

where

\[
\text{Vote}(p_i[x], y[x]) = \begin{cases} 
1 & \text{if } p_i[x] == y[x] \\
0 & \text{else,} 
\end{cases} \]  
\[(18)\]
IV. EXPERIMENT

A. DATASET AND EVALUATION METRICS

To verify our model, we performed the experiments on three publicly available sound event classification datasets: SECL_UMONS [11], Urbansound8K [10], and ESC-50 [9].

1) SECL_UMONS

SECL_UMONS\(^1\) is a real sound recording dataset for simultaneous classification and localization of sound events. The data samples in the SECL_UMONS are recording in two indoor room environments. The room condition of the first one is RT60 of 0.7s with the room dimensions of \(7.8 \times 3.6 \times 2.45\) [m] and second one is RT60 of 0.9s with room dimensions of \(9.4 \times 7.5 \times 4.85\) [m]. The dataset consists of 11 indoor event classes, such as chair_movement, cup_drop_off, furniture_drawer, hand_clap, keyboard, knock, phone_ring, radio, speaker, step, and whistle. The dataset contains 2178 sequences for the training set and 484 sequences for the validation set. We used single-channel waveform for our comparison. While the dataset offers classification label and localization label, but we only used classification label for our experiments. To evaluate the classification performance of the model and to compare to the baseline [11] system, we calculate F1-score. The F1-score is defined as

\[
\text{Precision} = \frac{TP}{TP + FP} \tag{19}
\]

\[
\text{Recall} = \frac{TP}{TP + FN} \tag{20}
\]

\[
F1 - \text{score} = \frac{2 \times \text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}} \tag{21}
\]

where TP is true positive, FP is false positive and FN is false negative samples.

2) Urbansound8K

Our outdoor dataset, Urbansound8K\(^2\), consists of 10 classes of urban sound recordings. The class of acoustic scenes are air_conditioner, car_horn, children_playing, dog_bark, drilling, engine_idling, gun_shot, jackhammer, siren, and street_music. The dataset contains 8732 labeled urban sound recordings. To compare to other state-of-the-art methods, 10-fold cross validation was used for accuracy calculation. The accuracy is defined as

\[
\text{ACC} = \frac{TP + TN}{TP + FP + FN + TN} \tag{22}
\]

where TN is true negative samples.

3) ESC-50

ESC-50\(^3\) [9] is used to evaluate our model using the dataset with a larger number of data classes. The dataset consists of 5-second long recordings organized into 50 semantical classes loosely arranged into 5 major categories:

\(^1\) https://zenodo.org/record/3632377#.YWTZQ9pByUk
\(^2\) https://urbansounddataset.weebly.com/urbansound8k.html
\(^3\) https://github.com/karolpiczak/ESC-50

animals, natural_soundscapes, non_speech_human_sounds, interior_sounds, and exterior_noises. The dataset contains 2000 labeled recordings with 40 examples per class. To verify the classification performance of the model, 5-fold cross validation was used for accuracy calculation.

4) CONFUSION MATRIX

We assess our model with the confusion matrix, which is a specific table layout that allows visualization of the classification results. TP, FP, FN, and TN can be counted using confusion matrix. TP is the number of samples that predict a class sample for that class. FP is the number of samples that predicted a non-class sample as that class. FN is the number of predicted samples of a class that are not of that class. TN is the number of samples that predicted a non-class sample as not of that class. When the class categories are more than two, we count TP, FP, FN, and TN and calculate precision, recall, accuracy and F1-score for each class, and average them to get final accuracy and F1-score. For the normalized confusion matrix, each column of the confusion matrix is normalized by the number of each class sample.

B. EXPERIMENT SETTINGS

We used Specmix [40] data augmentation strategy to increase the diversity of data distribution. Stochastic gradient descent was applied with a batch size 16, the momentum of 0.9, and weight decay \(1 \times 10^{-3}\) for regularization. Each network was trained for 310 epochs using a learning rate schedule with warm restarts that reset the learning rate to its maximum value of 0.1 after 10, 30, 70, 150 epochs, and then decays according to a cosine pattern to \(1 \times 10^{-5}\). This method helps performance improvement as verified by [17].

To find the optimal parameters and model, we used validation set in SECL_UMONS data set and cross-validation in Urbansound8K and ESC-50 data sets. For the SECL_UMONS, we trained the model with changing parameters, and assess them using validation set, which is randomly selected in training set on every epoch, and choose the best model. For the Urbansound8K, we separated the data set into 10 folds and selected one fold to test set and the rest of the folds to train set. We set 10 folds one by one as a test set and performed cross validation. For the ESC-50, we separated the data set into 5 folds and repeat the cross validation process.

C. COMPARISON OF THE PREVIOUS MODELS

1) INDOOR CONDITIONS

The results are shown in Table 1. In terms of F1-score, the proposed model outperforms the baseline model and the dual-path residual networks [17] by 1.94% and 3.01%, respectively. To evaluate the effectiveness of the feedback loop, compared to the proposed model with and without the feedback loop. The F1-score of the model with the feedback loop is 1.32% better than the model without the feedback loop. Therefore, the feedback implemented in our proposed model clearly improves sound event classification.
in indoor conditions. In Fig. 5(a), we provide the normalized confusion matrix yielded by the proposed model. All classes except `keyboard` class are classified over 95% accuracy. As a result, the proposed encoder module and feedback module are suitable to classify sounds recorded in indoor conditions.

2) OUTDOOR CONDITIONS
The results are shown in Table 2. Our model achieves 81.40% mean accuracy, which is the best result compared to state-of-the-art models. We also tested the effectiveness of the feedback loop in outdoor conditions the same as indoor conditions. The mean accuracy of the model with a feedback loop is 1.93% better than the model without a feedback loop. The proposed feedback loop, therefore, is equally effective in outdoor conditions. Fig. 5(b) shows the normalized confusion matrix yielded by the proposed model, which is the best result of the 10-fold validations.

3) THE LARGE NUMBER OF CLASSES
The results are shown in Table 3. Our model achieves 71.10% mean accuracy, which is not competitive result compared to state-of-the-art models. We assume that the cause of the bad result is the encoder structure. To verify this assumption and the effect of the feedback module in the ESC-50 data set, we designed a new model that consists of AST [21] encoder and feedback decoder. The feedback decoder, which consists of 5 fully connected layers with 4 feedback loops, is connected to the output layer of the AST model. As a result, the AST model with feedback module achieved 96.2% accuracy, which is 0.5% higher than the original AST model. Therefore, the feedback loops help to improve performance. However, there is a problem that the encoder structure must be well determined because the performance is greatly affected by the encoder structure.
To justify the effectiveness of the feedback module, output features of each feedback loop are visualized. We first extract the output features of each feedback loop for every validation data. After applying dimensional reduction to the extracted features, we checked how well the features were classified by comparing the reduced features with the class label. The visualization results are shown in Fig. 6. The feature distribution of the second, third, and fourth loop shows a better classification than the feature distribution of the first loop. Especially, orange class and turquoise class are confused in the first loop, but it seems to classify well in the third loop. Also, the distances between samples of different classes tends to increase.

### E. ABLATION STUDY

The proposed method contains some prominent technical proposals: input features, applying data augmentation, $\gamma$, feedback loop, and inference search strategy. We test the variation of the proposed model with some ablation studies using SECL_UMONS dataset.

1)  **EFFECT OF INPUT FEATURES**

We test the model using alternating $\text{nfft}$ and Mel-Spectrogram, delta, and delta-delta features to find a suitable input feature for the model. The results are shown in Table 3. The effect of $\text{nfft}$ was not great, and it was confirmed that the performance improved little by little each time del and del-del were added.

| Features       | F1-score(%) |
|----------------|-------------|
| nfft=128, mel  | 94.68       |
| nfft=128, mel + del | 95.07     |
| nfft=128, mel + del-del | 96.59    |
| nfft=256, mel   | 94.68       |
| nfft=256, mel + del | 94.75     |
| f=256, mel + del-del | 95.08   |

| Data augmentation | F1-score(%) |
|-------------------|-------------|
| None              | 94.21       |
| Mixup [44]        | 95.25       |
| Cutmix [45]       | 95.85       |
| SpecAugment [46] | 96.41       |
| SpecMix [40]      | 96.59       |

| $\gamma_1, \gamma_2, \gamma_3, \gamma_4$ | F1-score(%) |
|------------------------------------------|-------------|
| 1.0, 0.5, 0.1, 0.0                       | 94.26       |
| 0.5, 0.5, 0.5, 0.5                       | 96.59       |
| 0.0, 0.1, 0.5, 1.0                       | 96.41       |

2) **EFFECT OF DATA AUGMENTATION**

Since the proposed model should be trained well using generalized training data distribution, we test the model with various data augmentation strategies. We compare Mixup [44], Cutmix [45], Specaugment [46], and Specmix [40], which are data augmentation strategies for Spectrogram data. As a result, Specmix outperforms other strategies. In addition, the model was successfully trained for all data augmentation methods.

3) **EFFECT OF $\gamma$s**

To evaluate the effect of $\gamma$s, we test the model with different orders of $\gamma$s. The sort order of gamma changes the way it learns by changing the loss function. For example, the case that $\gamma$s are sorted by ascending order denotes that the mid-level features pass through the feedback loop, it sequentially solves from easy to difficult tasks, because of the characteristic of focal loss. We compare three cases: descending order, equal values, and ascending order. As a result, the case that $\gamma$s are equal value is better than other cases.

4) **EFFECT OF THE NUMBER OF FEEDBACK LOOP $N$**

To find the optimal number of feedback loop $N$, we test the models with changing $N$ in the range of 1 to 6. From the results, the performance gradually improves for $N = 1$ to 4 and then decreases from $N = 5$.

5) **EFFECT OF THE INFERENCE SEARCH**

We compare three different inference search algorithms. As a result, the maximum training accuracy method outperforms...
other methods. The best training model matches the best test model, demonstrating good generalization performance. However, contrary to the expectation that better features will be learned as it goes through the feedback loop, there are cases where the loop in the early stage performed better and is selected as the best training model. This instability is a future challenge.

V. CONCLUSION

In this study, we proposed a novel deep neural networks model using dual-path residual networks and feedback modules for sound event classification. Our method adopted weighted recurrent inference of sound event classification via the proposed linked feedback modules. Through the experiments, we demonstrated that the proposed model shows significant improvements in terms of sound event classification compared to state-of-the-art methods both in indoor and outdoor conditions. Additionally, the effectiveness of the feedback module was demonstrated with ablalion studies. We expected that the proposed sound event classification model can be used to abnormal detection, sound event detection, audio monitoring, healthcare, social robot, and scene understanding. Furthermore, the feedback module and recurrent inference methods can be applied to several tasks, such as classification, denoising, and generation. Moreover, the feedback module is easily adapted as a sub-module of the neural networks. A potential limitation, however, of the proposed model was that performance may vary greatly depending on the encoder structure. Future work will explore the efficacy of the feedback structure in solving other learning based tasks that may benefit weighted recurrent inference manner.

REFERENCES

[1] D. de Benito-Gorrón, D. Ramos, and D. T. Toledano, “A multi-resolution CRNN-based approach for semi-supervised sound event detection in DCASE 2020 challenge,” IEEE Access, vol. 9, pp. 89029–89042, 2021.

[2] E. Fonseca, M. Plakal, F. Font, D. P. W. Ellis, and X. Serra, “Audio tagging with noisy labels and minimal supervision,” 2019, arXiv:1906.02975.

[3] J. P. Bello, C. Silva, O. Nov, R. L. DuBois, A. Arora, J. Salamon, C. Mydlarz, and H. Doraiswamy, “SONYC: A system for the monitoring, analysis and mitigation of urban noise pollution,” 2018, arXiv:1805.06889.

[4] Y. Zigel, D. Litvak, and I. Gannot, “A method for automatic fall detection of elderly people using floor vibrations and sound—Proof of concept on human mimicking doll falls,” IEEE Trans. Biomed. Eng., vol. 56, no. 12, pp. 2858–2867, Dec. 2009.

[5] D. Stowell, D. Giannoulis, E. Benetos, M. Lagrange, and M. D. Plumbley, “Detection and classification of acoustic scenes and events,” IEEE Trans. Multimedia, vol. 17, no. 10, pp. 1733–1746, Oct. 2015.

[6] S. Mun, S. Park, D. K. Han, and H. Ko, “Generative adversarial network based acoustic scene training set augmentation and selection using SVM hyper-plane,” in Proc. DCASE, 2017, pp. 93–97.

[7] S. Suh, S. Park, Y. Jeong, and T. Lee, “Designing acoustic scene classification models with CNN variants,” DCASE2020 Challenge, Detection Classification Acoustic Scenes Events, Tech. Rep., Jun. 2020.

[8] W. Gao and M. McDonnell, “Acoustic scene classification using deep residual networks with focal loss and mild domain adaptation,” DCASE2020 Challenge, Detection Classification Acoustic Scenes Events, Tech. Rep., Jun. 2020.

[9] K. J. Piczak, “ESC: Dataset for environmental sound classification,” in Proc. 23rd ACM Int. Conf. Multimedia, Oct. 2015, pp. 1015–1018.

[10] J. Salamon, C. Jacoby, and J. P. Bello, “A dataset and taxonomy for urban sound research,” in Proc. 22nd ACM Int. Conf. Multimedia, Nov. 2014, pp. 1041–1044.

[11] M. Brousmiche, J. Rouat, and S. Dupont, “SECL-UMons database for sound event classification and localization,” in Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP), May 2020, pp. 756–760.

[12] A. Mesaros, T. Heittola, and T. Virtanen, “TUT database for acoustic scene classification and sound event detection,” in Proc. 24th Eur. Signal Process. Conf. (EUSIPCO), Aug. 2016, pp. 1128–1132.

[13] X. Bao, T. Gao, J. Du, and L.-R. Dai, “An investigation of high-resolution modeling units of deep neural networks for acoustic scene classification,” in Proc. Int. Joint Conf. Neural Netw. (IJCNN), May 2017, pp. 3028–3035.

[14] L. Nanni, Y. M. Costa, D. R. Lucio, C. N. Silla, and S. Brahnam, “Combining visual and acoustic features for audio classification tasks,” Pattern Recognit. Lett., vol. 88, pp. 49–56, Mar. 2017.

[15] K. J. Piczak, “Environmental sound classification with convolutional neural networks,” in Proc. IEEE 25th Int. Workshop Mach. Learn. Signal Process. (MLSP), Sep. 2015, pp. 1–6.

[16] J. Salamon and J. P. Bello, “Deep convolutional neural networks and data augmentation for environmental sound classification,” IEEE Signal Process. Lett., vol. 24, no. 3, pp. 279–283, Mar. 2017.

[17] M. D. McDonnell and W. Gao, “Acoustic scene classification using deep residual networks with late fusion of separated high and low frequency paths,” in Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP), May 2020, pp. 141–145.

[18] A. Kumar and V. Ithapu, “A sequential self-teaching approach for improving generalization in sound event recognition,” in Proc. Int. Conf. Mach. Learn., 2020, pp. 5447–5457.

[19] D. Kim, S. Park, D. K. Han, and H. Ko, “Multi-band CNN architecture using adaptive frequency filter for acoustic event classification,” Appl. Acoust., vol. 172, Jan. 2021, Art. no. 107579.

[20] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” in Proc. Adv. Neural Inf. Process. Syst., 2017, pp. 5998–6008.

[21] Y. Gong, Y.-A. Chung, and J. Glass, “AST: Audio spectrogram transformer,” 2021, arXiv:2104.01778.

[22] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “ImageNet: A large-scale hierarchical image database,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2009, pp. 248–255.

[23] M. Dorfer and G. Widmer, “Training general-purpose audio tagging networks with noisy labels and iterative self-verification,” in Proc. Detect. Classification Acoustic Scenes Events Workshop (DCASE), 2018, pp. 178–182.

[24] Y. Chen, Q. Guo, X. Liang, J. Wang, and Y. Qian, “Environmental sound classification with dilated convolutions,” Appl. Acoust., vol. 148, pp. 123–132, May 2019.

[25] F. Demir, M. Turkoglu, M. Aslan, and A. Sengur, “A new pyramidal concatenated CNN approach for environmental sound classification,” Appl. Acoust., vol. 170, Dec. 2020, Art. no. 107520.

[26] P. Putzky and M. Welling, “Recent inference machines for solving inverse problems,” 2017, arXiv:1706.04908.

[27] Z. Li, J. Yang, Z. Liu, X. Yang, G. Jeon, and W. Wu, “Feedback network for image super-resolution,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 3867–3876.

[28] J. Lee, J. Park, K. Lee, J. Min, G. Kim, B. Lee, B. Ku, D. K. Han, and H. Ko, “FBRNN: Feedback recurrent neural network for extreme image super-resolution,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW), Jun. 2020, pp. 488–489.

[29] J. Hupé, A. James, S. Payne, S. Lomber, P. Girard, and J. Bullier, “Cortical feedback improves discrimination between figure and background by V1, V2 and V3 neurons,” Nature, vol. 394, no. 6695, pp. 784–787, 1998.

[30] C. D. Gilbert and A. M. Sigman, “Brain states: Top-down influences in sensory processing,” Neuron, vol. 54, no. 5, pp. 677–696, 2007.

[31] M. Stollenga, J. Maso, F. Gomez, and J. Schmidhuber, “Deep networks with internal selective attention through feedback connections,” 2014, arXiv:1407.3068.
[32] A. R. Zamir, T.-L. Wu, L. Sun, W. B. Shen, B. E. Shi, J. Malik, and S. Savarese, “Feedback networks,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jul. 2017, pp. 1308–1317.

[33] H. T. Siegelmann and E. D. Sontag, “Turing computability with neural nets,” Appl. Math. Lett., vol. 4, no. 6, pp. 77–80, 1991.

[34] S. I. Mimilakis, K. Drossos, J. F. Santos, G. Schuller, T. Virtanen, and Y. Bengio, “Monaural singing voice separation with skip-filtering connections and recurrent inference of time-frequency mask,” in Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP), Apr. 2018, pp. 721–725.

[35] C. Zhu, Y. Zhao, S. Huang, K. Tu, and Y. Ma, “Structured attentions for visual question answering,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Oct. 2017, pp. 1291–1300.

[36] G. Song and W. Chai, “Collaborative learning for deep neural networks,” 2018, arXiv:1805.11761.

[37] M. D. McDonnell, “Training wide residual networks for deployment using a single bit for each weight,” 2018, arXiv:1802.08530.

[38] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, “Learning phrase representations using RNN encoder-decoder for statistical machine translation,” 2014, arXiv:1406.1078.

[39] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, “Focal loss for dense object detection,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Oct. 2017, pp. 2980–2988.

[40] G. Kim, D. K. Han, and H. Ko, “SpecMix: A mixed sample data augmentation method for training with time-frequency domain features,” 2021, arXiv:2108.03020.

[41] F. Medhat, D. Chesmore, and J. Robinson, “Masked conditional neural networks for environmental sound classification,” in Proc. Int. Conf. Innov. Techn. Appl. Artif. Intell. Cham, Switzerland: Springer, 2017, pp. 21–33.

[42] L. Nanni, G. Maguolo, S. Brahnam, and M. Paci, “An ensemble of convolutional neural networks for audio classification,” Appl. Sci., vol. 11, no. 13, p. 5796, Jun. 2021.

[43] J. Kim, “Urban sound tagging using multi-channel audio feature with convolutional neural networks,” in Proc. Detection Classification Acoustic Scenes Events, 2020, pp. 1–4.

[44] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, “Mixup: Beyond empirical risk minimization,” 2017, arXiv:1710.09412.

[45] S. Yun, D. Han, S. Chun, S. J. Oh, Y. Yoo, and J. Choe, “CutMix: Regularization strategy to train strong classifiers with localizable features,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2019, pp. 6023–6032.

[46] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, “SpecAugment: A simple data augmentation method for automatic speech recognition,” 2019, arXiv:1904.08779.