***Abstract*—Bioacoustics is an important area of study for its applications in conservation, biodiversity monitoring, and endangered species monitoring. As the use of deep neural networks for bioacoustical tasks continues to increase, developing methods for processing acoustical signals effectively is an important area of study. A key component to such processing is the effective elimination of noise from the acoustical signal. Recently, methods such as Per-channel Energy Normalization (PCEN) and Multi-rate PCEN (MRPCEN) have been implemented for this purpose. We propose a frontend based off of these processes called Trainable Multi-rate PCEN (TMRPCEN) that introduces a trainable functionality to the important parameters within these methods. Our frontend demonstrates improvement on both the PCEN and MRPCEN method, achieving an average accuracy of 85.23% and an averaged F1 score of 0.847 using 10-fold cross validation on a condensed dataset for bird detection. These results demonstrate the exciting potential of TMRPCEN for enhancing bioacoustical analysis for avian vocalizations and beyond.**

Index Terms-*Acoustic signal, acoustic noise, bioacoustics, bird sound, Multi-rate PCEN, Per-channel energy normalization, spectrogram.*

I. Introduction

A. Bioacoustics

Within the field of deep learning, there are a number of important areas of research involving audio processing. One such subfield is called bioacoustics.

Bioacoustics is a field that combines the biological and acoustical sciences, using technologies to collect, store, and analyze data related to animal sounds [1]. These sounds are typically communications between living organisms, often animals or insects, and generally take the form of vocalizations, such as with a bird call.

Spencer Perkins, Chang-Hsing Lee, C. C. Lien

With the use of passive acoustic monitoring (PAM), we can record the sound environment of a habitat and gain important insight into, amongst other [1] things, species diversity of the area, the presence of endangered species, and patterns in migration to that area. However, such data are time consuming for human annotation and analysis, and within the realm of acoustical signals, prone to human error. Due to this, the use of deep learning techniques for bioacoustical study has become more prominent, and has applications in areas such as conservation, biodiversity monitoring, and endangered species monitoring.

B. Bioacoustical Machine Listening

Taking cues from fields such as computer vision, deep neural networks (DNNs) have shown immense potential within the realm of machine listening. These deep learning architectures, often in the form of convolutional neural networks (CNNs) have emerged as standard tools for research on bioacoustical tasks, in addition to many of the other subfields of audio tasks.

One of the most essential steps within research in bioacoustical machine listening is the processing of the raw audio signal into a time-frequency representation that can then be passed into the CNN for analysis. These representations capture the spectral content of the audio signal over time. Due to this, it is possible to extract spectrotemporal patterns from the frequency contours [2] that are present within these time-frequency representations.

While such methods have undoubtedly aided in our ability to process and understand acoustical signals, thereby aiding in research within the field of bioacoustics, there are still challenges that are faced for such research. One important characteristic that is present within a bioacoustical task is the environment from which the audio signal is taken. Within a field setting, numerous challenges arise. Among them, the issue of environmental noise, including potential interference from the audio sensor itself, is a prominent standout. Such noises have the capacity to overshadow the target sound event, resulting in the target event being obscured from the foreground of the acoustical signal. Further, in an uncontrolled setting, as is a field setting, there is the possibility of many overlapping sound events, which could potentially obscure the target event.

Recently,

This study proposes TMRPCEN, a trainable frontend that expands on recent work done with Per-channel Energy Normalization (PCEN) and Multi-rate PCEN (MRPCEN) for bird sound detection.

The remainder of this paper is organized as follows: Section II introduces a standard audio featurization method, Log-Mel spectrogram, as well as more recent

developments, PCEN and MRPCEN, alongside this study's proposed TMRPCEN. Section III details the experimental setup for this study. Section IV presents the experimental results as well as the results of an ablation study. Finally, section V presents the conclusions drawn from this research.

II. Audio Feature Extraction

A. Log-Mel Spectrogram

An important step in any audio related task is the extraction of features from the raw audio data. Standard acoustical feature extraction often encompasses the following three steps:

1. A time-frequency representation of the audio signal is extracted, often by utilizing the short-time Fourier transform (STFT).
2. The filterbank is transformed to the Mel-scale.
3. Logarithmic compression is applied to the Mel-scale filterbank.

The second step in this process is a logarithmic transformation applied to the frequencies of the signal that is reminiscent of the way that humans perceive pitch. The third step is applied to reduce the dynamic range of the filterbank. The result of these three steps is a time-frequency representation known as a Log-Mel spectrogram. This representation is fairly standard in many deep learning methods for audio related tasks.

B. Per-Channel Energy Normalization

While Log-Mel spectrograms have proven to be effective in many audio tasks involving DNNs, they are not without limitations. In their study regarding a keyword spotting task, Wang et al. [3] proposed Per-Channel Energy Normalization (PCEN) as an alternative to logarithmic compression. PCEN was later applied to a bioacoustical event detection task by Lostanlen et al. [4] and has since displayed effectiveness in a range of audio tasks.

PCEN is a combination of adaptive gain control (AGC), first scaled by the impulse response of an autoregressive filter , followed by dynamic range compression (DRC) done on a spectrogram on a per-channel basis. The resulting equation is

, (1)

where controls the strength of AGC, is a small constant to avoid division by zero, and and are the parameters associated with dynamic range compression.

The original implementation of PCEN by Wang et al. noted that the parameters and are differentiable, and therefore updatable during backpropagation when training a DNN.

The scaling by the impulse response of is given by

where s is the smoothing coefficient or weight of the autoregressive filter and represents the hop size of the input spectrogram. As detailed by Lostanlen et al. [2], this results in a lowpass filter of 0dB gain. The cutoff frequency is defined by at 3 dB with a sidelobe falloff of 10 dB per decade near .

The overall effect of PCEN is a repression of noise while preserving acoustic events of importance. A comparison of the output of a logarithmic compression approach and PCEN approach are shown in Figure 1.

C. Multi-rate PCEN

An important step of PCEN is obtaining , the smoothed version of the input spectrogram, . The critical parameter in this process is s, the smoothing coefficient, which is directly related to the rate parameter, T, as it defines the cutoff frequency of the lowpass filter.

Ick and McFee [5] noted that the choice of T can be difficult in a field setting as the characteristics of different sound classes may vary greatly. Inspired by 3-channel RGB images, they proposed Multi-rate PCEN (MRPCEN) which varies the T parameter at each layer to create a multi-layered representation. This produces different levels of gain control at each layer, allowing that a sound event that may be suppressed in one layer will be is preserved in another. Figure 2 shows the output of PCEN at different T rates. Each output from this figure could represent a layer within MRPCEN at that T rate.

D. Trainable Multi-rate PCEN

This study proposes a frontend based off of PCEN and MRPCEN that utilizes learnable and parameters for AGC and DRC alongside multiple layers of trainable s parameters for the smoothing of the input spectrogram. We call this frontend TMRPCEN. Our method is based on two key factors: (1) the PCEN parameters and are differentiable and therefore updatable via backpropagation within a training algorithm; (2) MRPCEN can be broken down into a process of stacking layers of PCEN computations at different s rates, so it is possible to combine trainable PCEN with learnable s rates at each layer.

[70] In our method, the *s* parameters are initialized and updated on a per-channel basis at each layer corresponding to each rate within the TMRPCEN frontend. As mentioned, this is done in addition to learnable and parameters. The output of this frontend is then passed to a convolutional neural network (CNN). The CNN output is passed into a loss function and the trainable parameters are updated with the CNN's weights during backpropagation.

[71]

[72] III. Experimental Design

[73]

[74] *A. Dataset and Feature Extraction*

[75] This study uses a condensed version of the BirdVox-DCASE-20k dataset. The reader is referred to [6] for further information on the Bird-Vox-full-night dataset, from which this dataset was adapted. The original BirdVox-DCASE-20k dataset consists of 20,000 10-second audio clips recorded in a single night by six autonomous recording units. Evaluation methods for this dataset involve the detection of bird vocalizations within each clip without precise time stamps. Out of the 20,000 clips, 10,017 contain at least one bird vocalization, amounting to 50.09% of the dataset. During preliminary experiments, hardware issues made training on the full 20,000 samples unfeasible. To overcome this, we condensed the dataset by taking the first 2,000 samples from this dataset. The percentage of clips containing bird vocalizations in our condensed version is 50.55%. Our method utilizes 10-fold cross validation with each fold containing 200 audio clips.

[76]

[77] Initial feature extraction is done using Librosa 0.9.2 to generate our mel-spectrograms. The spectrograms were extracted with a sampling rate of 44.1 kHz, a window size of 1,024 samples, a hop length of 512 samples, and 128 mel-frequency bands. The result was 862 samples per 128 mel-frequency bands. These features were then used as input to our frontend models for experimentation.

[78]

[79] *B. Frontend and CNN Models*

[80] In addition to this study's proposed TMRPCEN, we also run comparison experiments on a logarithmic compression frontend, PCEN frontend, and MRPCEN frontend, both with learnable and parameters.

[81]

[82] For the Log-mel frontend, logarithmic compression was applied to the mel-spectrogram directly after computation using the Librosa 0.9.2 library utilized to produce the spectrogram. The logarithmic compression frontend can be seen as a baseline 0, as previous work has shown to improve upon this method by using PCEN. However, due to the continued use of such compression techniques across many audio tasks, we included this method in our research.

[83]

[84] The PCEN frontend is our primary baseline and was developed using Pytorch 1.13.1. It serves as a preliminary layer to the CNN, first computing the smoothed version of the input spectrogram, , followed by the application of AGC and DRC. The trainable AGC parameter, , and the DRC parameters, and , are initialized at 0.8, 10, and 0.25 respectively, following practical recommendations from previous research on bird sound tasks [2]. The *T* parameter is set at 2^5 resulting

[85] in . This value was chosen as a midway point between the ten logarithmically spaced *T* values used in the MRPCEN frontend. The parameter is set to 10e-6. The initializations for and , as well as the setting for remain constant for the PCEN, MRPCEN, and TMRPCEN experiments.

[86]

[87] Both the MRPCEN and TMRPCEN follow a similar design and flow, a forward pass of which can be seen in Figure 3. The MRPCEN *T* values are ten logarithmically spaced values ranging from 2^0-2^9 to remain consistent with the original implementation [5]. TMRPCEN initializes ten *s* values randomly from a normal distribution with mean 0.03 ($T\ 2^5$) and standard deviation 0.1. The outputs of each frontend are then passed to the CNN for bird detection.

[88]

[89] The CNN model employed for this study is based off the L^3 audio subnetwork [7] with some modifications. First, our initial convolutional block outputs 32 feature maps rather than 64, and our final

[90] convolutional block outputs 256 feature maps as opposed to 512. Secondly, we added two dense layers

[91] after our final convolutional block, both of which are followed by a dropout layer. These modifications were made during the preliminary tuning stage to address overfitting.

[92]

[93] Additionally, the input to the first layer is adjusted to support multiple rates (e.g. for the log compression and PCEN frontends, input is equal to 1; for the MRPCEN frontends, input is equal to 10). All layers utilize the Rectified Linear Units (ReLU) activation function except for the output layer, which uses the Sigmoid activation function for our binary

[94] classification task. Figure 4 displays the architecture of the CNN.

[95]

[96] *C. Training Algorithm*

[97] In each of our experiments, our frontend models and CNN are trained for 75 epochs. We use the binary cross entropy loss function for evaluating the predictions and the ADAM optimizer with a fixed learning rate of

0.0001 and weight decay of 0.001 for optimization. As stated, validation is done using 10-fold cross validation and our evaluation metrics are averaged accuracy and averaged F1 score across these ten folds. The training algorithm can be seen in Figure 5.

IV. Experimental Results and Ablation Study

The results of our experiments with these four frontends can be viewed in Table 1. It can be seen that our proposed TMRPCEN frontend improves upon the other three methods, achieving an averaged accuracy of 85.23% and an averaged F1 score of 0.847. Furthermore, the similarity in results from the PCEN and MRPCEN frontend highlight the importance of implementing the trainable *s* parameter alongside the learnable parameters and present in the PCEN and MRPCEN frontends.

In the original implementation of MRPCEN, Ick and McFee utilized ten logarithmically spaced *T* parameters. Similarly, TMRPCEN initializes ten *s* parameters. To investigate the effect of a larger or smaller number of *s* parameters, we performed an ablation study on our TMRPCEN frontend model, beginning with twelve *s* rates, and then reducing the number of trainable *s* parameters within each experiment. The results of this ablation study are shown in Table 2. Our original implementation with ten *s* rates remains the highest performer. Interestingly, the effect of reducing the number of rates from ten actually causing the frontend to underperform the PCEN frontend from the original experiments.

Further investigation into the effects of higher and lower number of rates, even across subfields of acoustical tasks may provide useful

insight into the most effective way to initialize multi-rate learnable *s* parameters.

V. Conclusions

Bioacoustical tasks within the realm of deep learning are an important and budding area of research. A common challenge within this subfield is suppression of unwanted noise within the input acoustical signal, allowing the model to correctly detect or classify the target event within. To this end, our research developed a trainable multi-rate PCEN

frontend called TMRPCEN. Our method expands on previous work with PCEN and MRPCEN, adding learnable *s* parameters at multiple layers to the initial smoothing process alongside the learnable adaptive gain control parameter and dynamic range compression parameters, and .

TMRPCEN improves upon these previous methods, in addition to the often utilized logarithmically compressed mel-spectrogram, achieving an averaged accuracy of 85.23% and averaged F1 score of 0.847 using 10-fold cross validation on a condensed version of the BirdVox-DCASE-20k dataset for bird detection. Such results demonstrate the potential of TMRPCEN in elevating bioacoustical analysis of avian vocalizations and more within the field of deep learning.

References

[1] W. Penar, A. Magiera, C. Klocek, "Applications of bioacoustics in animal ecology," *Ecological Complexity,* vol. 43, 2020, art. 100847.

[2] V. Lostanlen, J. Salamon, M. Cartwright, B. McFee, A. Farnsworth, S. Kelling, and J. P. Bello, "Per-channel energy normalization: Why and how," *IEEE Signal Processing Letters*, vol. 26, no. 1, pp. 39–43, 2019.

[3] Y. Wang, P. Getreuer, T. Hughes, R. Lyon, and R. Saurous, "Trainable frontend for robust and far-field keyword spotting," *ICASSP,* pp. 5670-5674, 2016.

[4] V. Lostanlen, J. Salamon, A. Farnsworth, S. Kelling, J.P. Bello, "Robust sound event detection in bioacoustics sensor networks," PLoS ONE, vol. 14, 2019.

[5] C. Ick and B. McFee, "Sound event detection in urban audio with single and multi-rate pcen,"*ICASS*, pp. 880-884, 2021.

[6] V. Lostanlen, J. Salamon, A. Farnsworth, S. Kelling, J. Bello. BirdVox-full-night: a dataset and benchmark for avian flight call detection, Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), 2018.

[7] R. Arandjelovic and A. Zisserman, "Look, listen and learn," *ICCV*, pp.609-617, 2017.

Spencer Perkins is with the Chung Hua University, 707, Sec.2, WuFu Rd., Hsinchu 30012, Taiwan (e-mail: spencer.perkins8a@gmail.com).

Chang-Hsing Lee is with Chung Hua University, 707, Sec.2, WuFu Rd., Hsinchu 30012, Taiwan.

C.C Lien is with Chung Hua University, 707, Sec.2, WuFu Rd., Hsinchu 30012, Taiwan.

[142]