ROBUST FEATURE LEARNING FOR ACOUSTIC SCENE CLASSIFICATION WITH MULTIPLE DEVICES

Technical Report

Yuzhong Wu, Tan Lee

The Chinese University of Hong Kong
Department of Electronic Engineering, Shatin, N.T., Hong Kong S.A.R., China
yzwu@link.cuhk.edu.hk, tanlee@cuhk.edu.hk

ABSTRACT

This technical report describes our submission for Task 1A of DCASE2020 challenge. The objective of the task is to identify acoustic scenes from audios recorded by various recording devices. In our ASC systems, we use sound-duration based decomposition method to decompose the time-frequency (TF) features into 3 components. Our observation shows that low frequency bins of the long-duration component image are most easily affected by the change of recording devices. We use an AlexNet-like CNN model with the decomposed TF features to build ASC systems. To prevent the CNN classifier from over-fitting to the seen recording devices in the training dataset, we apply an auxiliary classifier on the embedding feature extracted from long-duration component image. We propose the regularized cross-entropy (RCE) loss to train the auxiliary classifier. Experiment results on development dataset shows that the use of regularized cross-entropy loss significantly improves the CNN accuracy on audios from unseen devices.

Index Terms— Acoustic scene classification, convolutional neural network, feature decomposition, regularized cross-entropy loss

1. INTRODUCTION

Acoustic scene classification (ASC) is the task of classifying recorded audio signal into one of predefined scene classes. It has been one of the major task in IEEE AASP Challenge on Detection and Classification of Acoustic Scenes and Events (DCASE) since 2013. This report describes the details of our submission for task 1A of DCASE 2020. The objective of the task is to build ASC systems with generalization ability across various recording devices.

In this report, we use the wavelet-based filter-bank (scalogram) features to construct our ASC systems. The scalogram features are further decomposed into 3 component images using the method described in [1]. Observation on spectrum mapping coefficients from different devices to device A shows that the low frequency bins are most affected by the difference of recording devices. Besides, when scalogram features are decomposed, we observe that most of the device-relevant acoustic information lies in S_{long} component which contains the long-lasting background sounds. Thus, we propose several ASC systems which are more robust to the change of recording devices compared to an AlexNet-like CNN baseline.

To further prevent the CNN classifier from over-fitting to the seen recording devices in the training data, we apply an auxiliary classifier on the embedding feature extracted from long-duration component image. Instead of training with plain cross-entropy loss, we propose a novel loss function called regularized cross-entropy (RCE) loss for training auxiliary classifier. We formulate the RCE loss as the weighted combination of the CE loss and a regularization loss, which serves as a strong regularizer to make the CNN classifier be less focus on the long-lasting background patterns for ASC, and thus prevent the classifier from being bias to the audios from seen recording devices. Experimental results on development dataset show that our systems significantly improve the ASC accuracy towards audio signals from unseen devices.

2. FEATURE DESIGN

2.1. Wavelet-Based Filter-Bank Features

We use wavelet-based filter-bank features in our proposed ASC models. Previous works have shown that applying wavelet filter-bank on STFT results in a better feature representation than applying mel filter-bank for ASC task [1, 2]. We follow the same setting as [1] to extract the scalogram features. Figure 1 shows the 128 wavelet filters used to extract the scalogram features. Compared with log-mel features with the same number of frequency bins, the scalogram features have relatively higher frequency resolution in low frequencies.

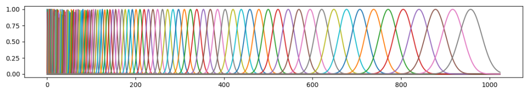

Figure 1: The wavelet filters used to extract the saclogram features. The x-axis represents the frequency points in short-time Fourier transform (STFT).

2.2. Time-Frequency Feature Decomposition

The method of time-frequency feature decomposition is based on median filtering. It was studied initially with log-mel features in [3] and further investigated in [1]. Experimental results show that CNNs trained with decomposed time-frequency features perform better than CNNs trained with un-decomposed features. The decomposition method allows the separation of long-lasting background sound information and transient sound information, which enables more fine-grained analysis of acoustic scene signals.

In this report, scalogram feature S is decomposed into 3 component images: S_{long}, S_{medium} and S_{short}. S_{long} contains the long-duration sounds, S_{median} contains the medium-duration sounds and S_{short} contains short-duration sounds. Notice that the following relationship holds:

$$S = S_{long} + S_{medium} + S_{short}. \tag{1}$$

3. ANALYSIS ON RECORDING DEVICES

Acoustic scene signals recorded with different devices may have different frequency response. The frequency response refers to the output level of recording device over its receivable frequency range.

3.1. Frequency Response Modeling

We assume the frequency response of a recording device to be linear time-invariant (LTI). Denote two recording devices "A" and "B", we model the relationship between an audio signal $x_A[n]$ (recorded by device "A") and a parallel audio signal $x_B[n]$ (recorded by device "B") in Fourier domain as:

$$|X_A[f]| = exp(C_{B,A}) \cdot |X_B[f]|, \tag{2}$$

where $exp(\cdot)$ is the exponential function and $C_{B,A}$ is a vector with real constant values used to map the frequency intensities of X_B to device X_A. Notice that it is common to represent frequency intensity in logarithm scale:

$$log(|X_A[f]|) = C_{B,A} + log(|X_B[f]|), \tag{3}$$

and we name the values of $C_{B,A}$ as the spectrum mapping coefficients.

3.2. Frequency Response of Multiple Devices

The TAU Urban Acoustic Scenes 2020 Mobile development dataset [4] contains audios from 9 devices. In the training set there are audios from 6 devices: A, B, C, S1, S2, S3. After extraction of scalogram features as described in Section 2.1, we calculate the spectrum mapping coefficients from different devices to device A using parallel audios. Then we take the mean of the coefficients across all time frames and all audio signals. The results are shown as in Figure 2. It can be observed from the figure that the spectrum mapping coefficients are most diverge in low frequency bins.

Spectrum mapping coefficients for the decomposed scalogram features are also considered. After decomposing the scalogram features S into 3 component images (S_{long}, S_{medium} and S_{short}) using two median filters with kernel size 201 and 11, we plot the spectrum mapping coefficients from various devices to device A for each component image as in Figure 3. As we can see, the spectrum mapping coefficients for S_{medium} and S_{short} are very close to 0. The difference caused by recording devices mainly lies in S_{long}. Thus, specifically making S_{long} to be device-invariant is the key to construct robust feature representation for ASC with various devices.

4. PROPOSED ASC SYSTEMS

4.1. Baseline System Design

The baseline system uses a CNN classifier whose architecture is given as in Table 1 (with the number of input channel n being 1). To make prediction on a 10-second audio from the dataset, the scalogram feature are extracted from the audio. Then it is cut into 1.28-second non-overlapping feature segments. Each segment is fed into the CNN classifier to obtain the segment-level soft predictions. The soft prediction on the 10-second audio is obtained by averaging the soft predictions of its segments.

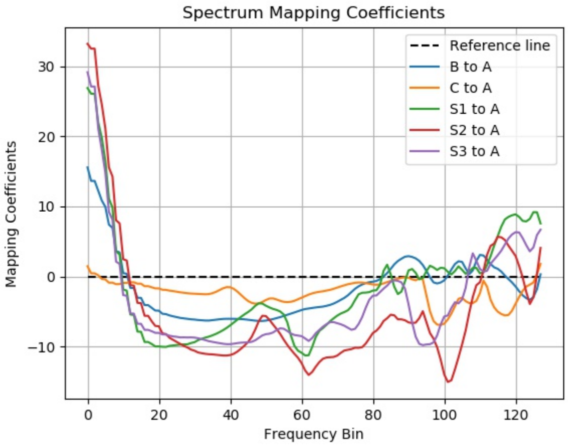

Figure 2: Mean spectrum mapping coefficients from different devices to device A. The coefficients are calculated using scalogram features as described in Section 2.1.

Table 1: The CNN architecture used to construct our ASC system. n is the number of input channels.

1	Input $n \times 128 \times 128$
2	3x3 Convolution-BN-ReLU (48 filters)
3	2x2 Max Pooling
4	3x3 Convolution-BN-ReLU (96 filters)
5	2x2 Max Pooling
6	3x3 Convolution-BN-ReLU (192 filters)
7	2x2 Max Pooling
8	3x3 Convolution-BN-ReLU (192 filters)
9	3x3 Convolution-BN-ReLU (192 filters)
10	2x2 Max Pooling
11	Flattening
12	Fully Connected (dim-1024)-BN-ReLU
13	Fully Connected (dim-256)-BN-ReLU
14	10-way Sigmoid

4.2. Robust ASC Systems for Various Devices

We propose multiple single-model ASC systems to deal with the classifier bias towards seen recording devices. They are listed as follows:

- System A is similar to the baseline model. The only difference is that, instead of using the plain scalogram features, we apply the 1-D filter $[-1, 0, +1]$ on the time axis for each frequency bin of the scalogram features. In this way we remove the long-lasting background sounds and only the transient sounds are preserved.

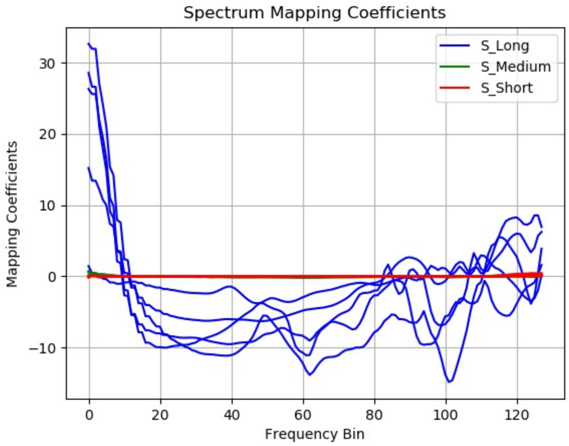

Figure 3: Mean spectrum mapping coefficients from various devices to device A calculated using decomposed scalogram feature component S_{long}, S_{medium} and S_{short}. There are $5 \times 3 = 15$ curves in this figure (some curves are overlapping). The coefficient curves computed from the same component image are shown in same color.

- System B uses the decomposed scalogram features. The scalogram features are decomposed using two median filters with kernel size 201 and 11. As each time frame represent 0.01 second, after decomposition S_{long} contains sounds longer than 1 second, S_{median} contains sounds with duration being in the range from 1 second to 0.05 second, and S_{short} contains sounds shorter than 0.05 second. Given $n = 3$, we still use the CNN architecture in Table 1, but the convolutional layers are divided into 3 groups. This is to guide the CNN model to learn long-lasting background sounds, medium duration sounds and transient sounds separately.
- System C is a modification of System B. We discard the S_{long} in System B and thus the number of input channels $n = 2$. The convolutional layers of CNN model are divided into 2 groups. The reason of discarding S_{long} is because it is very sensitive to the change of recording devices, as illustrated in Section 3.2.
- System D is a also a modification of System B. Instead of discard the whole S_{long}, we only discard some of the frequency bins which are easily affected by the change of recording devices (by setting them to zeros after input normalization). We empirically select the first 15 bins and the last 36 bins to discard.
- System E further considers the manipulation of S_{long}. In this system we use all component images. To prevent the model from over-fitting to the seen devices because of S_{long}, an auxiliary scene classifier with input being the time-axis-averaged embedding feature of S_{long} is trained with regularized cross-entropy (RCE) loss. Figure 4 illustrate the system design for System E. The RCE loss is a weighted sum of the CE loss and a regularization loss, which will be described in Section 5.

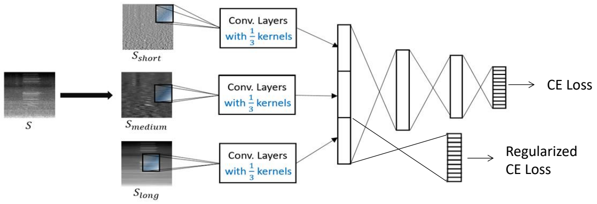
S_{short}
S
S_{medium}
S_{long}
Conv. Layers with $\frac{1}{3}$ kernels
Conv. Layers with $\frac{1}{3}$ kernels
Conv. Layers with $\frac{1}{3}$ kernels
CE Loss
Regularized CE Loss

Figure 4: System E with an auxiliary classifier with input being embeddings of S_{long} trained with regularized cross-entropy (RCE) loss.

5. REGULARIZED CROSS-ENTROPY LOSS

5.1. Cross-Entropy Loss

The cross-entropy loss is widely used in training deep classification models. Under a binary classification scenario, denote $y = 0$ or 1 as the ground-truth label, $p \in (0, 1)$ as the output probability of the DNN classifier, cross entropy (CE) loss can be defined as:

$$L_{CE} = \begin{cases} -log(p) & \text{if y=1} \\ -log(1-p) & \text{if y=0.} \end{cases} \tag{4}$$

For multi-class classification problem, the following binary cross-entropy loss is calculated for each class:

$$L_{BCE}(p) = -y \cdot log(p) - (1-y) \cdot log(1-p). \tag{5}$$

We use the sigmoid function $\sigma(\cdot)$ as the output activation function of our CNN models. In this case, the output probability of CNN is obtained by applying sigmoid function to logit x:

$$p = \sigma(x) \tag{6}$$

5.2. Regularized Cross-Entropy Loss

The formulation of regularized cross-entropy (RCE) loss is inspired by focal loss [5], starting from the concept of easy/hard samples. An easy sample means a training sample with high predicted probability on ground-truth class, while a hard sample means a training sample with low predicted probability on ground-truth class. Considering the gradient of plain CE loss, the gradient magnitude is large for hard samples (predicted probability < 0.5) and is small for easy samples (predicted probability > 0.5). Focal loss is basically making the gradient magnitude for hard samples to be larger (by making the loss curve to be steeper), and thus it can work well if we encounter severe class imbalance during model training. On the other hand, if we have many outliers in the training data, we may hope to do the opposite of focal loss – to reduce the learning weights of hard samples. Specifically in ASC task, we want the CNN classifier not to learn too much details from S_{long} as it is sensitive to the change of recording devices. Thus, the RCE loss is proposed for CNN training.

The RCE loss is a combination of two loss terms: the CE loss L_{CE} and a regularization loss L_R. The regularization loss is designed as a symmetric version of CE loss, which has large gradient magnitude for easy samples and small gradient magnitude for hard samples (outliers). The RCE loss L_{RCE} is given by:

$$L_{RCE} = (1-\alpha)L_{CE} + \alpha L_R \tag{7}$$

with

$$L_R = \begin{cases} log(1-p) & \text{if y=1} \\ log(p) & \text{if y=0.} \end{cases} \quad (8)$$

Notice that $\alpha \in [0, 1]$ is a weighting parameter to control the degree of regularization. If $\alpha = 0$, the RCE loss becomes the plain CE loss. If $\alpha = 0.5$, then the loss is a linear function of logit x. The loss curve of L_{RCE} for different values of α is shown as in Figure 5. Notice that the loss value can be negative, however for model learning it is the gradient of the loss that truly matters.

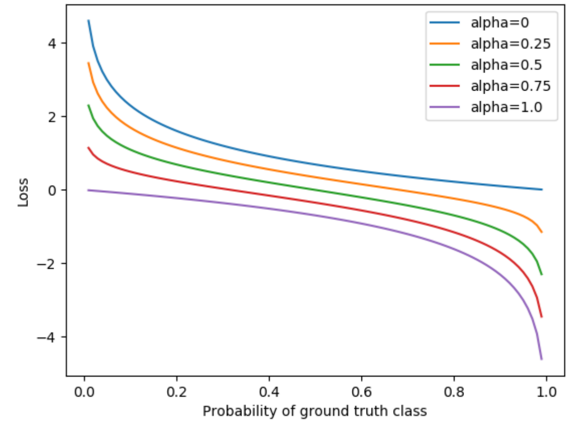

Figure 5: The regularized cross-entropy loss with different α.

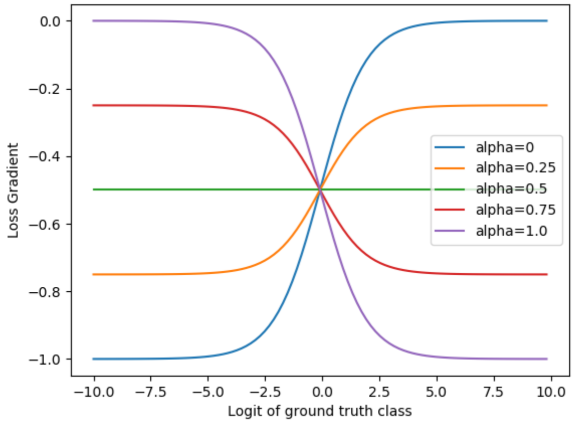

Figure 6: The gradient curve of RCE loss w.r.t. logit (sigmoid is used). When $\alpha = 0$ the RCE loss becomes the plain CE loss.

6. EXPERIMENTS

6.1. Data Preprocessing

The TAU Urban Acoustic Scenes 2020 Mobile development dataset [4] is used for model training and testing. For each 10-second binaural audio signal in the dataset, STFT is applied on audio waveform with 2048 FFT points, window length of 25 ms and hop length of 10 ms. Wavelet filter-bank is applied on the logarithm magnitude of the STFT result to obtain the scalogram features. The resulted scalogram feature has the shape $(1000, 128)$ where 1000 is the number of time frames and 128 is the number of frequency bins.

6.2. Optimization

We use initial learning rate of 0.0001, and the learning rate is multiplied with 0.5 after every 4 epoch. The number of training epochs is 40. Adam optimizer [6] ($\beta_1 = 0.9$ and $\beta_2 = 0.999$) is used. Weight decay with coefficient 0.0015 is used for regularization purpose. Mixup [7] approach is used for data augmentation.

6.3. Results and Discussion

Table 2 shows the performance single-model ASC systems described in Section 4. From the table it can be seen that: (1) Compared to the baseline system, System A has an improved accuracy for unseen devices, while the accuracy for seen devices drops a little bit. (2) Using decomposed scalogram features (System B) performs better than the baseline system for both seen and unseen devices. (3) Discarding the S_{long} component (System C) raises the accuracy for unseen devices but decreases the accuracy for seen device. (4) Discarding some device-sensitive frequency bins in S_{long} (System D) can maintain most of the accuracy for seen devices. However, the improvement of accuracy on unseen devices is limited compared to System C. (5) Using RCE loss ($\alpha = 1$) trained on auxiliary classifier with input being embeddings of S_{long} (System E) achieves the best accuracy for unseen devices, while preserving most of the accuracy for seen devices. Notice that $\alpha = 1$ may not be an optimal choice, and the accuracy can be further improved by tuning α.

Table 2: Accuracy of single-model ASC systems on the development dataset. Description of systems is in Section 4.

	Seen Device	Unseen Device	Overall
Baseline	65.7%	46.0%	59.1%
System A	62.8%	59.2%	61.6%
System B	**67.9**%	49.9%	61.9%
System C	64.1%	60.2%	62.8%
System D	67.6%	51.6%	62.3%
System E	66.0%	**61.0**%	**64.3**%

7. SYSTEM SUBMISSION

We have 4 system submissions for the Task 1A of DCASE 2020 challenge. The systems are trained using the entire development dataset. The first submission (Wu_CUHK_task1a_1) is a single-model ASC system which uses System E with $\alpha = 0.75$. The second submission (Wu_CUHK_task1a_2) is an ensemble of System A, B, C and D. The third submission (Wu_CUHK_task1a_3) is an ensemble of five independently trained System E with α being 0, 0.25, 0.5, 0.75 and 1.0 respectively. The fourth submission (Wu_CUHK_task1a_4) is an ensemble of the third submission's models and System A, B, C, D.

8. REFERENCES

[1] Y. Wu and T. Lee, “Time-frequency feature decomposition based on sound duration for acoustic scene classification,” in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 716–720.

[2] H. Chen, Z. Liu, Z. Liu, P. Zhang, and Y. Yan, “Integrating the data augmentation scheme with various classifiers for acoustic scene modeling,” DCASE2019 Challenge, Tech. Rep., June 2019.

[3] Y. Wu and T. Lee, “Stratified time-frequency features for CNN-based acoustic scene classification,” DCASE2019 Challenge, Tech. Rep., June 2019.

[4] T. Heittola, A. Mesaros, and T. Virtanen. (2020, Feb.) TAU Urban Acoustic Scenes 2020 Mobile, Development dataset. [Online]. Available: https://doi.org/10.5281/zenodo.3819968

[5] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, “Focal Loss for Dense Object Detection,” *arXiv e-prints*, p. arXiv:1708.02002, Aug. 2017.

[6] D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization,” *arXiv e-prints*, p. arXiv:1412.6980, Dec. 2014.

[7] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, “mixup: Beyond Empirical Risk Minimization,” *arXiv e-prints*, p. arXiv:1710.09412, Oct 2017.