AN ACOUSTIC SCENE CLASSIFICATION SYSTEM FOR RESOLVING RECORDING ENVIRONMENT OFFSETS

Yongpeng Yan, Wuyang Liu, Yi Chai

Whuhan University

ABSTRACT

In this technical report, we present a system to tackle the ICME2024 Grand Challenge, which is to solve the domain shift problem in ASC between different city. We propose a system to address this problem, using ResNet as the backbone network. We enhance domain generalization capability by optimizing the information bottleneck problem and maximizing feature entropy, we employ the MixMatch framework for semi-supervised training.

***Index Terms*—** Acoustic scene classification, Domain generalization, ResNet, self-supervised learning

1. INTRODUCTION

Acoustic scene classification(ASC) is a task in the field of audio signal processing and machine learning, where the goal is to classify or categorize audio recordings based on the type or scene they represent. The objective is to automatically recognize and label the environmental context or scene from which an audio clip originates.

Domain generalization(DG) is a machine learning problem that aims to build models that can generalize well to unseen or novel domains. In traditional machine learning, models are typically trained and evaluated on data that is assumed to be drawn from the same distribution as the test data. However, in real-world scenarios, the test data may come from different domains with variations in data distribution, characteristics, or conditions.

This task is aim to solve the problem of sound scene domain migration caused by the change of the city.In the following text, we provided a brief overview of our submitted system. The main architecture of our model is ResNet, and we utilized the semi-supervised method called MixMatch. We incorporated three different approaches to tackle the domain shift issue.

2. PROPOSED METHOD

2.1. Semi-supervised training method

We have followed the same approach used by mixmatch [1], which was a semi-supervised learning method first proposed in the field of CV and has achieved excellent results in multiple datasets. The specific method is to first take a batch of labeled data and a batch of unlabeled data, and perform k data augmentation operations on the unlabeled data to obtain X and kU.

The unlabeled data is then fed into the classifier to obtain the output predicted by the classifier, noting that the gradient is not calculated in this step. Next, the average classification probability of all unlabeled data is calculated, the Temperature Sharpen algorithm is applied to obtain the guess label q of the unlabeled data, and finally X and kU are mixed together and randomly rearranged to obtain a new dataset $\mathcal{W}$.

Then divide $\mathcal{W}$ into two parts, the first part is the same size as X, and the second part is the same size as kU, which are denoted as W_x and W_u respectively, and then we act mixup on W_x and X, W_u and U. Finally we get X' and U' to calculate the classifier loss.

The loss function contains two parts, the labelled data loss L_x and unlabelled dataloss L_u, show as:

$$L_x = \frac{1}{|X'|} \sum_{x,p \in X'} H(p, P_{model}(y|x;\theta)) \tag{1}$$

$$L_u = \frac{1}{L|U'|} \sum_{u,q \in U'} \|q; P_{model}(y|u;\theta)\|_2^2 \tag{2}$$

$$L = L_x + \lambda_U \cdot L_u \tag{3}$$

2.2. Model

We compared the performance of three basic model architectures in this task: CNN, ResNet, and Transformer. We adopted the CNN and ResNet networks from [2], and for the Transformer, we chose HTS-AT [3] as the backbone network. After multiple rounds of experiments and comparisons, we found that ResNet achieved the highest accuracy and generalization performance. Therefore, we selected ResNet as the model to be submitted.

ResNet [4] model is a residual network, which has 17 convolutional layers. There is no frequency subsampling throughout the whole network. Each input feature map is divided into two sub-feature maps along the frequency dimension. To be specific, if we have N frequency bins, the first N/2 and the second half are processed by two parallel stacked

convolutional layers. Thus, we have a two-stage model structure. At last, a global pooling layer and 10-way softmax are used to get the final utterance level prediction results.

2.3. Data augmentation method

In the MixMatch semi-supervised learning method, we only used data augmentation on unlabeled data. In this approach, we apply perturbations to the data through data augmentation k times and then use the consistency principle to constrain them. We employed three types of data augmentation methods: time masking, frequency masking, and random time shifting. For time masking and frequency masking, we utilized the library functions TimeMasking and FrequencyMasking provided by PyTorch. As for random time shifting, we first sample $\lambda \in (0, 1)$, and then cut the Mel spectrogram alone time dimention with length of λT. Them we shift the two part and connect them into a new Mel spectrogram of length T.

2.4. Domain generalization method

We conducted three approaches to address the domain generalization problem.

Firstly, in the input samples, for the labeled features, we randomly applied mixup by extracting randomly selected data with the same label from the training set. Since the data originates from various cities, this operation helps in mitigating the differences in domain distribution.

Secondly, we observed that the classification targets, which consist of 10 sound scenes, can be classified into three major categories: outdoor, indoor, and vehicle. Furthermore, the model often exhibited classification errors within these three major categories during testing. To address this issue, we incorporated a three-class classification loss into the overall loss function. Specifically, for the labeled data, we computed the model's output and summed the logits for the sub-classes within each major class. Then, we calculated the cross-entropy loss by comparing the summed logits with the one-hot labels representing the three-class classification.

Thirdly, at the feature level, we incorporated two additional losses following the method described in [5]. These losses are the classwise instance discrimination (CID) loss and the feature dimension correlation (FDC) loss.

The CID loss is utilized to address the information bottleneck problem by encouraging the model to learn more informative representations. It achieves this by differentiating between instances of the same class. The CID loss is calculated at the classification feature level as:

$$\ell_i = -\log \frac{\sum_{z_i^+ \in \{class\, i\}, z_i^+ \neq z_i} \exp((z_i^T z_i^+)/\tau)}{\sum_{j=1, j\neq i}^{N} \exp((z_i^T z_j)/\tau)} \quad (4)$$

$$L_{CID} = \sum_{i=1}^{N} \ell_i \quad (5)$$

The FDC loss aims to maximize feature entropy and foster the exploration of diverse and discriminative features. It captures the correlation between different dimensions of the features and promotes high-dimensional diversity. Similar to the CID loss, the FDC loss is computed at the classification feature level as:

$$G_{d,d'} = \sum_{b=1}^{B} \widetilde{Z}_{b,d} \cdot \widetilde{Z}_{b,d'} \quad (6)$$

$$L_{FDC} = \sum_{i=1}^{D} \sum_{j=1, j\neq i}^{D} G_{ij}^2 \quad (7)$$

3. EXPERIMENTAL SETUP AND RESULTS

3.1. Dataset and feature extraction

We performed pre-training on the TAU UAS 2020 Mobile development dataset, which was provided in the Decase2020 Task 1. This dataset consists of 7 classes that are the same as the classes in our task, and all the audio samples in this dataset have a length of 10 seconds. We processed the data by using log-mel spectrogram features along with their first-order and second-order differences as inputs to the model. The specific parameters we used for the feature extraction are as follows: 128 mel bands, 2048 FFT signal length, 2048 window size, and 1024 hop length.The final input feature size to the model is [batch, 3, 128, 423].

3.2. Training setup and results

During the pre-training phase, the training parameters for our model are as follows: the learning rate is fixed at 1e-3, the batch size is 16, and the training lasts for 20 epochs. In the CAS dataset, the parameters for the optimizer are as follows: the initial learning rate is 1e-3, with a step size of 2 and a gamma value of 0.9 for exponential decay. The batch size is 16, and each cycle consists of 200 iterations.For the data augmentation parameters, we set the parameters for time masking and frequency masking to 200 and 20, respectively.In the MixMatch method, we adopted the original parameters from the method, which is set as T=0.5, α=0.75, and λ_u=75.

According to the baseline's dataset splitting strategy, we set aside 20% of the development dataset as the validation set. On the validation set, the model achieved a maximum accuracy of 100%. However, this is not the version we submitted.

Our final submission was selected based on a comparison of the classification loss and domain generalization (DG) loss at each epoch.

4. REFERENCES

[1] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin Raffel, "MixMatch: A Holistic Approach to Semi-Supervised Learning," *arXiv e-prints*, p. arXiv:1905.02249, May 2019.

[2] Hu Hu, Chao-Han Huck Yang, Xianjun Xia, Xue Bai, Xin Tang, Yajian Wang, Shutong Niu, Li Chai, Juanjuan Li, Hongning Zhu, Feng Bao, Yuanjun Zhao, Sabato Marco Siniscalchi, Yannan Wang, Jun Du, and Chin-Hui Lee, "Device-Robust Acoustic Scene Classification Based on Two-Stage Categorization and Data Augmentation," *arXiv e-prints*, p. arXiv:2007.08389, July 2020.

[3] Ke Chen, Xingjian Du, Bilei Zhu, Zejun Ma, Taylor Berg-Kirkpatrick, and Shlomo Dubnov, "HTS-AT: A Hierarchical Token-Semantic Audio Transformer for Sound Classification and Detection," *arXiv e-prints*, p. arXiv:2202.00874, Feb. 2022.

[4] Mark D. McDonnell and Wei Gao, "Acoustic scene classification using deep residual networks with late fusion of separated high and low frequency paths," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 141–145.

[5] Jiao Zhang, Xu-Yao Zhang, Chuang Wang, and Cheng-Lin Liu, "Deep representation learning for domain generalization with information bottleneck principle," *Pattern Recognition*, vol. 143, pp. 109737, 2023.