Wider or Deeper Neural Network Architecture for Acoustic Scene Classification with Mismatched Recording Devices

Lam Pham  
Lam.Pham@uit.ac.at  
Austrian Institute of Technology  
Austria

Khoa Tran  
tdkhoa@dut.undn.vn  
Da Nang University  
Viet Nam

Dat Ngo  
dn22678@essex.ac.uk  
University of Essex  
UK

Hieu Tang  
hieutq10@fpt.edu.vn  
FPT University  
Viet Nam

Son Phan  
son.pl@amanotes.com  
Amanotes Company  
Viet Nam

Alexander Schindler  
Alexander.Schindler@ait.ac.at  
Austrian Institute of Technology  
Austria

ABSTRACT

In this paper, we present a robust and low complexity model for Acoustic Scene Classification (ASC), the task of identifying the scene of an audio recording. We firstly construct an ASC model in which a novel inception-residual-based network architecture is proposed to deal with the issue of mismatched recording devices. To further improve the model performance but still satisfy the low footprint, we apply two techniques of ensemble of multiple spectrograms and model compression to the proposed ASC model. By conducting extensive experiments on the benchmark DCASE 2020 Task 1A Development dataset, we achieve the best model performing an accuracy of 71.3% and a low complexity of 0.5 Million (M) trainable parameters, which is very competitive to the state-of-the-art systems and potential for real-life applications on edge devices.

KEYWORDS

Deep learning, convolutional neural network (CNN), acoustic scene classification (ASC), data augmentation, model complexity, inception.

ACM Reference Format:
Lam Pham, Khoa Tran, Dat Ngo, Hieu Tang, Son Phan, and Alexander Schindler. 2022. Wider or Deeper Neural Network Architecture for Acoustic Scene Classification with Mismatched Recording Devices. In International Conference on Multimedia (ACM Multimedia Asia 2022), December 13–16, 2022, Tokyo, Japan. ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3551626.3564962

1 INTRODUCTION

The Acoustic Scene Classification (ASC) task, one of main topics in ‘Machine Hearing’ research field [14], has attracted much research attention recently. Indeed, not only more and more ASC datasets such as Lits Rouen [32], ESC50 [30], DCASE Task 1 [3], or Crowded Scenes [26] have been published, but various ASC systems, leveraging deep neural networks, have been also proposed (i.e. The literature review section in [22] summarizes state-of-the-art ASC systems as well as updated machine learning and deep learning techniques applied for ASC).

Regarding ASC challenges, they mainly come from different noise resources, various sounds in real-world environments, occurring as single sounds, continuous sounds or overlapping sounds, or dynamic energy of sound events in a sound scene recording. These challenges drive ASC research community to focus on analyzing frequency bands [11, 17, 29] rather than specific sound events [33]. However, the new issue of mismatched recording devices firstly mentioned in DCASE 2018 Task 1B challenge [3] further increases ASC challenge as this issue causes energy distribution at certain frequency bands of spectrograms from the same class significantly different (i.e. In Figure 1 of [31], Mel-based spectrograms from the same sound scene of ‘on Tram’ show different as they are from three different recording devices). To deal with the mismatched recording devices, ensemble of different spectrogram inputs [18, 19, 23–25, 27, 28] or ensemble of multiple classification models [5, 20] are mainly approached. However, ensemble methods present large footprint models, which is challenging to implement on edge devices or real-time applications.

This paper aims at developing an ASC model which is not only robust to deal with ASC challenges mentioned recently but also presents a low complexity with less than 1M parameters. To this end, we firstly construct an ASC model in which a novel neural network, a shallow and wide inception-residual-based architecture, is presented. The proposed ASC model is then compared with other deep learning based models using benchmark architectures such as VGGish networks (e.g., VGG16, VGG19) or residual based architectures (e.g., Resnet, DenseNet, MobileNet, or Xception) to evaluate whether a wider and shallow network or a deeper architecture is effective for ASC, specially with the mismatched recording device issue. We then apply two techniques: (1) ensemble of multiple spectrograms and (2) model compression to the proposed model,
achieve a low footprint ASC model but still perform robust and competitive to state-of-the-art systems.

2 THE PROPOSED ASC SYSTEM

We firstly construct our ASC model which presents a high-level architecture in Figure 1. As Figure 1 shows, the proposed ASC model can be separated into three main steps: The front-end feature extraction, the online data augmentations, and the back-end classification.

2.1 The front-end feature extraction

The audio recordings are firstly re-sampled to 32,000 Hz. Then, they are transformed into log-Mel spectrograms using Librosa [15]. By setting Hann window size, the hop size, the filter number to 2048, 1024, 128, respectively and applying delta, delta-delta on each spectrogram, we generate a log-Mel spectrogram of \(128\times305\times3\) from one 10-second audio segment. Notably, the channel dimension is 3, which causes by concatenating the original log-Mel spectrogram, delta, and delta-delta.

2.2 The online data augmentations

In this paper, we apply three data augmentation methods of Random Cropping [35], Specaugment [21], and Mixup [36, 37], respectively. In particular, the temporal dimension of log-Mel spectrograms of \(128\times305\times3\) is randomly cropped to \(128\times256\times3\) (e.g. Random Cropping method). Then, ten continuous and random frequency or temporal bins of the cropped spectrograms are erased (e.g. Specaugment method). Finally, the spectrograms are randomly mixed together using different ratios from Uniform or Beta distributions (e.g. Mixup method). All of three data augmentation methods are applied on each batch of spectrograms during the training process, referred to as the online data augmentations.

2.3 The back-end classification

As Figure 2 shows, the proposed back-end classification can be separated into two main parts: CNN-based deep neural network backbone and multilayer perceptron (MLP) based classification. In particular, the proposed CNN-based backbone comprises four blocks: one Inception Block and three Inc-Res Blocks as described at the upper part of Figure 2, which makes use of inception-based (e.g., Inception Block) or both inception-based and residual architectures (e.g., three Inc-Res Blocks). Three Inc-Res Blocks share the same network architecture, but channel numbers increases from 128, to 256 at two final Inc-Res Blocks. Four blocks of the CNN-based backbone are performed by Inception layers (Inc01[Channel] in Inception Block, Inc02[Channel×Kernel Size] in Inc-Res Blocks), Convolutional layer (Conv[Channel×Kernel Size]), Batch Normalization (BN) [7], Dropout (Dr(Drop Ratio) [34], Rectified Linear Unit (ReLU) [16], Max Pooling (MP[Kernel Size]), Average Pooling (AP[Kernel Size]), Residual Normalization (RN(\(\lambda = 0.4\))) inspired from [8]).

Regarding two Inc01 layers used in Inception Block as shown in the left part of Figure 2, we use fixed kernel sizes of \([3\times3],[1\times1]\), and \([4\times1]\) (Note that using the kernel \([4\times1]\) helps to focus on frequency bands). Meanwhile, kernel sizes used in Inc02 layers in three Inc-Res Blocks as shown in the right part of Figure 2 are defined by kernel size \(K\). By using different kernel sizes of \([K\times1],[K\times K]\), and \([1\times K]\), then applying AP layers with the same kernels, and finally adding output of these AP layers together, the network can learn the distribution of energy in certain frequency bands effectively, which strengthens the network to tackle the issue of mismatched recording devices.

The MLP-based classification as shown in the lower part of Figure 2 performs a Pooling Block and two fully connected layer blocks. At Pooling Block, we extract three types of features from: (1) global average pooling across the channel dimension, (2) global max pooling across temporal dimension, and (3) global average pooling across frequency dimensions. We then concatenate these features before feeding into fully connected blocks. While the first fully connected layer (FC[Channel]) combines with ReLU and Dr, the second fully connected layer uses Softmax layer for classifying into \(C = 10\) scene categories.

To further evaluate whether a wider or deeper neural network architecture is effective for ASC with the issue of mismatched recording devices, we replace the proposed CNN-based backbone by different benchmark network architectures of VGG16, VGG19, MobileNetV1, MobileNetV2, ResNet50V2, ResNet101V2, ResNet152V2, DenseNet121, DenseNet169, DenseNet201, and Xception which are available from Keras Application API [1]. In other words, only the layers before the global pooling layer of these benchmark networks are used. These reused layers are then connected with the MLP-based classification of the proposed ASC model to perform end-to-end network architectures. These network architectures
We then fuse the probability results by using PROD late fusion. In particular, we evaluate this ensemble strategy in our paper. In particular, we use three spectrograms of log-Mel, Constant Q Transform (CQT) [15], and Gammatone filter (Gam) [2]. By using the same settings mentioned in Section 2.1, all spectrograms present the same size of $128 \times 305 \times 3$. For each type of spectrogram, we apply the same data augmentation methods mentioned in Section 2.2 and the proposed model presented in Section 2.3 for classification, referred to as CQT-model, log-Mel-model, and Gam-model, respectively. We then fuse the probability results by using PROD late fusion. In particular, we conduct experiments over individual network with different spectrogram inputs, then obtain predicted probability of each network as $\bar{p}_s = (\hat{p}_1, \hat{p}_2, ..., \hat{p}_C)$, where $C$ is the category number and the $s^{th}$ out of $S$ networks evaluated. Next, the predicted probability after PROD fusion $\bar{p}_{prod} = (\hat{p}_1, \hat{p}_2, ..., \hat{p}_C)$ is obtained by:

$$\bar{p}_s = \frac{1}{S} \prod_{s=1}^{S} \hat{p}_{sc} \text{ for } 1 \leq s \leq S$$  \hspace{1cm} (1)$$

Finally, the predicted label $\hat{y}$ is determined by

$$\hat{y} = \text{argmax}(\hat{p}_1, \hat{p}_2, ..., \hat{p}_C)$$  \hspace{1cm} (2)$$

### 4 EXPERIMENTS AND DISCUSSION

#### 4.1 Dataset and Evaluation Metric

**DCASE 2020 Task 1A Development set** [6]: The dataset comprises 23040 segments (duration of each is 10 seconds) with a total recording time of 64 hours. The dataset was recorded from three real devices namely A, B, and C with 40 hours, 3 hours, and 3 hours, respectively. Additionally, synthesized audio recordings namely from S1 to S6 with 3-hour recording time for each are added. As audio recordings are from both real and synthesized devices, this dataset is ideal to evaluate ASC task with the issue of mismatched recording devices.

We follow DCASE challenges, then separate the DCASE 2020 Task 1A Development set into Training and Evaluating subsets for training and evaluating processes, respectively (Note that audio recordings from S4, S5, and S6 are not presented in Training subset recordings and are used in evaluating processes). Additionally, synthesized audio recordings namely from S1 to S6 with 3-hour recording time for each are added. As audio recordings are from both real and synthesized devices, this dataset is ideal to evaluate ASC task with the issue of mismatched recording devices.

#### 4.2 Model Implementation

As using the Mixup data augmentation method, labels are not one-hot encoding format. Therefore, we use Kullback–Leibler divergence (KL) loss [13] shown in Eq. (3) below.

$$\text{Loss}_{KL}(\Theta) = \sum_{n=1}^{N} y_n \log \left( \frac{y_n}{\hat{y}_n} \right) + \frac{\lambda}{2} \sum_{i,j} |\Theta_{ij}|^2$$  \hspace{1cm} (3)$$

where $\Theta$ are trainable parameters, constant $\lambda$ is set initially to 0.0001, $N$ is batch size set to 100, $y_i$ and $\hat{y}_i$ denote expected and predicted results. We construct and train deep learning networks proposed with Tensorflow. We set epoch number=100 and using Adam method [10] for optimization. While a learning rate of 0.0001 is set for the first 80 epochs with data augmentation methods, a low learning rate of 0.00001 is set for the next 20 epochs without any data augmentation method.
4.3 Experimental results and discussion

As experimental results on DCASE 2020 Task 1A dataset are shown in Table 2, our proposed ASC model outperforms benchmark network architectures across recording devices. Further analyze performance of benchmark network architectures, it indicates that deeper neural networks such as VGG19, ResNet152V2 or DenseNet201 present low performance than the lower complexity networks such as VGG16, ResNet50V2, or DenseNet121 from the same architecture groups. This proves that a wider and shallow neural network is present low performance than the lower complexity networks such as VGG19, ResNet152V2 or DenseNet201.

Although applying model compression techniques helps to significantly reduce the model complexity, it affects the accuracy performance of single models as shown in Table 3. By using both ensemble of multiple spectrograms and model compression techniques (e.g., channel deconvolution and channel reduction), we can achieve ASC models which show a balance between the accuracy performance and the model complexity. Indeed, ensembles of three spectrograms using Red03 and Red04 achieve 71.3% with 0.5M and 70.9% with 0.3M respectively, which satisfies the target low footprint model.

4.4 Conclusion

This paper has presented a novel inception-residual-based neural network for ASC task with mismatched recording devices. By conducting intensive experiments over the benchmark DCASE 2020 Task 1A Development dataset, it is indicated that the novel network presenting a wider and shallower architecture is more effective for ASC rather than deeper architectures. Additionally, our proposed ensemble of multiple spectrograms and model compression (e.g., Red03) help to achieve an accuracy of 71.3% and low footprint of 0.5M trainable parameters, which shows a balance between the model performance and the model complexity. These results also prove that our proposed ASC models are competitive to the state-of-the-art systems and validates ASC application on edge devices.

Table 2: Compare the proposed ASC model to DCASE baseline and benchmark neural networks

| Performances | DCASE Baseline | Proposed Model | MobileV1 | MobileV2 | VGG16 | VGG19 | ResNet50V2 | ResNet152V2 | DenseNet121 | DenseNet201 | Vception |
|--------------|----------------|----------------|----------|----------|-------|-------|-----------|-------------|-------------|-------------|---------|
| A(%)         | 70.6           | 77.3           | 74.2     | 71.6     | 68.3  | 67.1  | 74.1      | 74.0         | 74.1        | 74.8        | 75.2    |
| B(%)         | 60.6           | 70.5           | 60.1     | 56.3     | 54.5  | 56.1  | 57.9      | 60.8         | 63.1        | 58.3        | 62.0    |
| C(%)         | 62.6           | 75.7           | 63.7     | 60.6     | 61.5  | 61.5  | 63.7      | 67.8         | 63.7        | 68.6        | 68.4    |
| S1(%)        | 55.0           | 69.7           | 57.2     | 52.5     | 55.3  | 49.8  | 68.2      | 52.5         | 62.0        | 57.2        | 60.1    |
| S2(%)        | 53.3           | 70.6           | 51.4     | 55.2     | 54.4  | 51.4  | 54.1      | 52.8         | 58.9        | 56.3        | 54.7    |
| S3(%)        | 51.7           | 71.8           | 55.4     | 52.5     | 53.5  | 52.0  | 55.6      | 57.0         | 60.2        | 59.6        | 62.2    |
| unseen-S4(%) | 48.2           | 61.5           | 43.8     | 41.3     | 43.8  | 38.3  | 45.6      | 47.6         | 51.7        | 51.5        | 50.4    |
| unseen-S5(%) | 45.2           | 66.1           | 44.7     | 46.1     | 45.3  | 44.4  | 52.0      | 44.3         | 53.8        | 48.7        | 49.4    |
| Average(%)   | 54.1           | 69.1           | 53.3     | 51.6     | 53.3  | 50.8  | 55.1      | 54.0         | 58.7        | 56.7        | 57.9    |
| Parameters(M) | 5.6            | 4.3            | 4.3      | 3.5      | 15.3  | 20.6  | 25.7      | 26.5         | 8.1         | 20.3        | 23.0    |
| Memory(MB)   | 19.2           | 16.6           | 16.4     | 13.7     | 58.2  | 254.8 | 98.0      | 230.6        | 30.9        | 77.5        | 87.6    |

Table 3: Performance comparison among single and ensemble models with or without model compression

| Single Models | Acc.(%) | Parameters (M) |
|---------------|---------|----------------|
| CQT-model     | 60.8    | 4.3            |
| CQT-model w/ Red01 | 58.7    | 1.6            |
| CQT-model w/ Red02 | 60.2    | 0.46           |
| CQT-model w/ Red03 | 61.0    | 0.17           |
| CQT-model w/ Red04 | 58.2    | 0.1            |
| Gam-model     | 65.8    | 4.3            |
| Gam-model w/ Red01 | 65.3    | 1.6            |
| Gam-model w/ Red02 | 64.3    | 0.46           |
| Gam-model w/ Red03 | 63.7    | 0.17           |
| Gam-model w/ Red04 | 61.9    | 0.1            |
| log-Mel-model | 69.1    | 4.3            |
| log-Mel-model w/ Red01 | 67.3    | 1.6            |
| log-Mel-model w/ Red02 | 67.4    | 0.46           |
| log-Mel-model w/ Red03 | 64.2    | 0.17           |
| log-Mel-model w/ Red04 | 65.6    | 0.1            |
| Ensemble Models | Acc.(%) | Parameters (M) |
| CQT, log-Mel, Gam-models | 73.6    | 12.9           |
| CQT, log-Mel, Gam-models w/ Red01 | 72.9    | 4.8            |
| CQT, log-Mel, Gam-models w/ Red02 | 72.8    | 1.4            |
| CQT, log-Mel, Gam-models w/ Red03 | 71.3    | 0.5            |
| CQT, log-Mel, Gam-models w/ Red04 | 70.9    | 0.3            |

Table 4: Compare our proposed best models to 5 best systems from DCASE 2020 Task 1A challenge

| Top-5 models [4] | Acc.(%) | Parameters (M) |
|------------------|---------|----------------|
| Top-5 (ensemble) | 73.6    | 12.9           |
| Top-4 (ensemble) | 72.9    | 4.8            |
| Top-3 (ensemble) | 72.0    | 1.4            |
| Top-2 (ensemble) | 71.3    | 0.5            |
| Top-1 (ensemble) | 70.9    | 0.3            |

Table 4 compares our best performance models with the top-five systems submitted to DCASE 2020 Task 1A challenge [3]. Our proposed model (e.g. the ensemble of CQT-model, Gam-model, and log-Mel-model) achieves the top-4 ranking which records an accuracy of 73.6% and present lower model footprint.

5 CONCLUSION

This paper has presented a novel inception-residual-based neural network for ASC task with mismatched recording devices. By conducting intensive experiments over the benchmark DCASE 2020 Task 1A Development dataset, it is indicated that the novel network presenting a wider and shallower architecture is more effective for ASC rather than deeper architectures. Additionally, our proposed ensemble of multiple spectrograms and model compression (e.g., Red03) help to achieve an accuracy of 71.3% and low footprint of 0.5M trainable parameters, which shows a balance between the model performance and the model complexity. These results also prove that our proposed ASC models are competitive to the state-of-the-art systems and validates ASC application on edge devices.

ACKNOWLEDGMENTS

The work described in this paper is performed in the H2020 project STARLIGHT (“Sustainable Autonomy and Resilience for LEAs using AI against High priority Threats”). This project has received funding from the European Union’s Horizon 2020 research and innovation program under grant agreement No 101021797.
