WAVELET BASED MEL SCALED REPRESENTATION FOR LOW COMPLEXITY ASC WITH MULTIPLE DEVICES

Technical Report

Aswathy Madhu

Department of Electronics & Communication
College of Engineering
Thiruvananthapuram, Kerala, India
aswathymadhu@cet.ac.in

Suresh K

Department of Electronics& Communication
Govt. Engineering College, Barton Hill
Thiruvananthapuram, Kerala, India
sureshk@cet.ac.in

ABSTRACT

This technical report presents our submission to the IEEE AASP Challenge on Detection and Classification of Acoustic Scenes and Events (DCASE) 2021 for Task1 (Acoustic Scene Classification), subtask A (Low-Complexity Acoustic Scene Classification with Multiple Devices). The proposed system is a simple state-of-the-art approach employing wavelet based mel scaled representation for acoustic signals and a CNN classifier. We use data augmentation to handle device mismatch and post training quantization of network weights to enforce low complexity in terms of model size. The submitted system surpasses the baseline system utilizing CNN developed for this subtask.

Index Terms— Acoustic Scene Classification, Convolution Neural Network, Deep Learning, biorthogonal wavelet

1. INTRODUCTION

Acoustic Scene Classification (ASC) [1] is aimed at associating a semantic label to an acoustic signal to characterize the environment in which it was recorded. This research field has been progressively growing in the last few years due to its enormous application potential in context aware devices and in intelligent monitoring devices [2]. This is evident from the increased participation in the events like IEEE AASP Challenge on Detection and Classification of Acoustic Scenes and Events (DCASE). With the emergence of deep learning algorithms, ASC systems have seen drastic improvement in accuracy. However, the focus of ASC has shifted from improvement in accuracy to incorporating real world scenarios.

Two prime concerns in real world scenarios are generalization capability and computational limitations. In the real world, it is sensible to assume that any ASC system will have audio samples from a large variety of recording devices. Additionally, for deployment in context aware devices which are usually computationally limited, the ASC system should take into account the computational complexity. This year, DCASE addresses these two concerns by introducing a subtask seeking a low complexity solution for ASC with multiple devices (Task 1 sub task A). As a solution to this sub task, we propose a simple state-of-the-art CNN based approach which utilizes wavelet based mel scaled representation for acoustic signals to facilitate learning. We address the issue of generalization across multiple devices with data augmentation and ensure low complexity in terms of model size with post training quantization of network weights.

The rest of the report is organized as follows. In section 2, we briefly describe the dataset employed in the task along with the signal representation technique. Then, we describe in detail the architecture of the CNN used along with the activation function and loss function used in the training process. Next, the implementation details of data augmentation methods and post training quantization are provided. Section 3 presents the results obtained and the associated analyses. Section 4 concludes the report.

2. METHOD

2.1. Dataset

The development dataset used in this task is TAU Urban Acoustic Scenes 2020 Mobile [3], development dataset. It contains recordings pertaining to 10 different acoustic scenes - *airport, bus, metro, metro station, park, public square, shopping mall, street pedestrian, street traffic* and *tram*. The development dataset contains recordings from 10 European cities. Three real devices (A, B, and C) and six simulated devices (S1 - S6) were used to record data. The developers have provided a training/test split of 70/30 in which 13965 samples are used for training and 2970 samples are used for testing.

2.2. Signal Representation

The defacto baseline for signal representation in deep ASC models is the mel spectrogram. The mel spectrogram representation has a fixed resolution in time and frequency since its computation is based on Short Time Fourier Transform (STFT). However, weak and multifaceted signals like acoustic scene signals can be better characterized by a transformation having non uniform resolution in time and frequency. Hence, we use a variant of mel spectrogram which is obtained by merging DWT with mel spectrogram for signal representation. We performed 4 level decomposition of the audio signals with a biorthogonal wavelet [4] (bior1.3) and computed the mel spectrogram of the resulting approximation and detail components. The mel spectrograms were computed for analysis frames of 40ms length with 50% overlap at 40 mel bands. The magnitudes of the mel spectrograms are scaled logarithmically. The final input to the CNN has the shape $40 \times 250 \times 2$. We use librosa [5] to process the audio signals.

2.3. Model Architecture

We use a simple CNN for classification. The model architecture is discussed below.

1. The first layer uses 32 swish activated 3 × 3 kernels with batch normalization.
2. The second layer has same settings as the first layer. Max pooling is applied with a pool size of (5,5) for dimensionality reduction. Dropout is introduced at a rate of 0.2.
3. The third layer has same settings as the second layer with the exception that it uses 64 kernels and the pool size is (4,100).
4. The fourth layer is a fully connected layer with 100 swish activated hidden units. Dropout is introduced at a rate of 0.2.
5. The final layer is the output layer with 10 output units and softmax activation.

The model is implemented in Keras [6]. For training, we use categorical focal loss as the loss function. The model is trained in batches of size 64 for 500 epochs with data shuffling between epochs. To select the best performing model, a validation set is used and the log loss and accuracy are computed after each epoch. The model parameters are optimized by Adam optimizer with default configurations.

2.4. Swish activation function

The choice of activation function has a large impact on the model performance. Over the last few years, ReLU has been the most popular activation function due to its simplicity and effectiveness. Recently, a new activation function has been proposed - swish [7]. This activation can formally be defined as

$$f(x) = x * (1 + exp(-x))^{-1} \tag{1}$$

Like ReLU, swish is bounded below and unbounded above. This property helps swish to offer strong regularization. Unlike ReLU, swish is smooth. This helps swish to optimize and generalize neural network. Additionally, swish is non monotonic which helps to improve the gradient flow. Hence, we adopt swish as the activation function for our model.

2.5. Categorical Focal loss

The focal loss [8] generalizes the binary and multi class cross entropy loss and is mathematically defined as:

$$FL(p_t) = -\alpha_t(1 - p_t)^{\gamma} log(p_t) \tag{2}$$

where α is the balancing parameter, γ is the focusing parameter and p_t is the prediction score. The higher the value of γ the lower the loss for well classified examples. Thus it is possible to turn the attention of the model towards hard-to-classify examples. α is used to balance the importance of positive/negative examples and it does not differentiate between easy/hard examples. We used $\alpha = 0.25$ and $\gamma = 2$ in our model.

2.6. Data Augmentation for generalization

In order to make the network extract the most appropriate features and to enhance the performance towards unseen data, we used data augmentation. Four elementary augmentation techniques as suggested in [9] are implemented along with MixUp [10] augmentation. We give the details of the augmentation techniques used in this work below.

1. Time stretching (TS): speed up or slow down the audio sample without altering pitch. We implemented time stretching using librosa.effects with a factor of 0.85.

2. Pitch shifting (PS1, PS2): change the pitch of the audio recording without altering duration. We implemented pitch shifting librosa.effects with two factors -2 and -1.5.

3. Additive background noise (BG): We added a background noise to each audio recording using the equation $y = x_1 + w.x_2$ where x_1 is the original audio recording, x_2 is the background noise recording and w is a weighting constant whose value is randomly chosen from a uniform distribution between 0.1 and 0.5.

4. Dynamic range compression (DRC): To compress the dynamic range of an audio, either the loud sounds are limited or the quiet sounds are enhanced. In this work, the dynamic range of audio was compressed to the speech profile by using SoX (Sound eXchange).

5. MixUp: MixUp was implemented using the equations $\tilde{x} = \lambda x_i + (1-\lambda)x_j$ and $\tilde{y} = \lambda y_i + (1-\lambda)y_j$ where x_i and x_j are raw input vectors and y_i and y_j are one hot label encodings. We used $\lambda = 0.2$ in our model.

2.7. Post training quantization

We reduced the model size by quantizing our trained model in Keras while converting it into TensorFlow Lite format using the TensorFlow Lite Converter. We adopted float16 quantization for the model weights since it reduces the model size by up to half (since all weights become half of their original size) while causing minimal loss in accuracy.

3. RESULTS AND DISCUSSIONS

We implemented DCASE 2021 Task 1A baseline in Keras to facilitate comparison. For evaluation of the model we used macro-average multi class cross-entropy (Log loss) and classification accuracy.

3.1. Performance of Baseline

Tables 1 and 2 shows the class-wise and device-wise results of the baseline on the development dataset. The overall performance of the baseline system on the development set is 43.7%. It is evident from the results that the baseline system has no explicit mechanism for handling the device mismatch. Also, the device-wise system performance varies according to the amount of training data.

Scene Label	Log loss	Accuracy
airport	1.483	45.6
bus	1.522	41.8
metro	1.572	44.8
metro station	1.454	45.8
park	1.527	42.8
public square	1.488	47.5
shopping mall	1.56	41.1
street pedestrian	1.47	47.1
street traffic	1.572	40.7
tram	1.545	39.9
Average	1.519	43.7

Table 1: Classwise performance of baseline

Device	Log loss	Accuracy
A	1.47	46.7
B	1.595	40.7
C	1.578	42.6
S1	1.501	40.3
S2	1.493	47.3
S3	1.47	43.3
S4	1.51	43
S5	1.584	45.2
S6	1.474	44.2

Table 2: Devicewise performance of baseline

3.2. Performance of Proposed Model

Tables 3 and 4 shows the class-wise and device-wise results of the proposed model on the development dataset. The overall performance of the proposed model on the development set is 85.1%. It is evident from the results that the proposed model significantly outperforms the baseline model (confirmed by a one-tailed z-test [11] $p < 0.01$). Also, the device-wise system performance is significantly enhanced due to data augmentation.

Scene Label	Log loss	Accuracy
airport	0.619	87.2
bus	0.692	80.8
metro	0.66	86.2
metro station	0.623	85.2
park	0.59	85.9
public square	0.62	85.2
shopping mall	0.6	84.8
street pedestrian	0.591	86.5
street traffic	0.653	84.2
tram	0.63	84.5
Average	**0.628**	**85.1**

Table 3: Classwise performance of proposed model

Device	Log loss	Accuracy
A	0.616	85.5
B	0.618	82.4
C	0.614	88.1
S1	0.623	84.2
S2	0.603	85.2
S3	0.651	86.1
S4	0.608	86.7
S5	0.656	83.9
S6	0.641	83.3

Table 4: Devicewise performance of proposed model

3.3. Model Complexity

To evaluate the proposed system from the computational load perspective, Table 5 provides the total number of parameters and complexity in terms of size for each layer of the proposed model. It is evident from the Table that the proposed model has a total of 42774 parameters of which 42518 are trainable and 256 are non trainable. The model has a size of 167.1 KB. However, after managing complexity with post training quantization of network weights to float16, the model size is reduced to 89.5 KB.

Layer	Parameters	Size
conv2d	608	2.375 KB
batch_norm	128	512 bytes
activation	0	0 bytes
conv2d_1	9248	36.12 KB
batch_norm_1	128	512 bytes
activation_1	0	0 bytes
max_pooling2d	0	0 bytes
dropout	0	0 bytes
conv2d_2	18496	72.25 KB
batch_norm_2	256	1 KB
activation_2	0	0 bytes
max_pooling2d_1	0	0 bytes
dropout_1	0	0 bytes
flatten	0	0 bytes
dense	12900	50.39 KB
activation_3	0	0 bytes
dropout_2	0	0 bytes
dense_1	1010	3.945 KB
activation_4	0	0 bytes
Total	**42774**	**167.1 KB**

Table 5: Model Summary

4. CONCLUSION

In this technical report, we presented a low complexity cross device ASC system utilizing wavelet based mel spectrogram representation.We handled the generalization problem with five simple data augmentation techniques and managed complexity consraints with post training quantization of network weights to float16 format. The proposed system was able to achieve 85.1% classification accuracy on the development dataset with a log loss of 0.628. The model size is 89.5 KB. These results show that the proposed model significantly outperforms the baseline as confirmed by a one tailed z-test with $p < 0.01$.

5. REFERENCES

[1] D. Barchiesi, D. Giannoulis, D. Stowell, and M. D. Plumbley, “Acoustic scene classification: Classifying environments from the sounds they produce,” *IEEE Signal Processing Magazine*, vol. 32, no. 3, pp. 16–34, 2015.

[2] L. Ma, D. Smith, and B. Milner, “Environmental noise classification for context-aware applications,” in *In Proc. EuroSpeech-2003*, 2003, pp. 2237–2240.

[3] A. Mesaros, T. Heittola, and T. Virtanen, “A multi-device dataset for urban acoustic scene classification,” in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2018 Workshop (DCASE2018)*, November 2018, pp. 9–13. [Online]. Available: https://arxiv.org/abs/1807.09840

[4] S. Mallat, *A wavelet tour of signal processing (2. ed.).* Academic Press, 1999.

[5] B. Mcfee, C. Raffel, D. Liang, D. P. Ellis, M. Mcvicar, and E. Battenberg, “librosa: Audio and music signal analysis in python,” *Proceedings of the 14th python in science conference*, 2015.

[6] F. Chollet *et al.*, “Keras,” 2015. [Online]. Available: https://github.com/fchollet/keras

[7] P. Ramachandran, B. Zoph, and Q. V. Le, “Swish: a Self-Gated Activation Function,” *ArXiv e-prints*, Oct. 2017.

[8] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, “Focal loss for dense object detection,” in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2999–3007.

[9] J. Salamon and J. P. Bello, “Deep Convolutional Neural Networks and Data Augmentation for Environmental Sound Classification,” *IEEE Signal Processing Letters*, vol. 24, no. 3, pp. 279–283, 2017. [Online]. Available: https://dx.doi.org/10.1109/lsp.2017.2657381

[10] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, “mixup: Beyond Empirical Risk Minimization,” in *International Conference on Learning Representations*, 2018. [Online]. Available: https://openreview.net/forum?id=r1Ddp1-Rb

[11] J. Schiller, R. A. Srinivasan, and M. Spiegel, *Schaum’s Outline of Probability and Statistics, 4th Edition*. US: McGraw-Hill, 2012. [Online]. Available: https://mhebooklibrary.com/doi/book/10.1036/9780071795586