Technical Paper: Deep Scattering Spectrum with Mobile Network for Low Complexity Acoustic Scene Classification

Xing Yong Kek

Newcastle University
Faculty of Science, Agriculture & Engineering
Singapore
x.y.kek2@newcastle.ac.uk

Cheng Siong Chin

Newcastle University
Faculty of Science, Agriculture & Engineering
Singapore
cheng.chin@newcastle.ac.uk

Ye Li

Xylem Water Solutions Singapore Pte Ltd
Decision Science
Singapore
ye.li@xylem.com

ABSTRACT

We present a technical paper that provide details of our classification model submitted to DCASE 2021 Task1a challenge. In this paper, we proposed the use of DSS with mobile network to tackle low complexity computation.

Index Terms— deep scattering spectrum, convolution neural network, low complexity computation, mobile network

1. INTRODUCTION

This paper provides a technical breakdown of our proposed model to tackle DCASE 2021 Task 1a [1,2], which requires the network model to be of low computational complexity on top of the ability for the model to identify it vicinity, given an acoustic recording. The use case of low computational complexity model is in line with the advancement and mass adaptation of Internet of Things and robotics, where on-node detection is required to make near real-time decision. Hence, low latency and less memory model is desired. There are applications that can benefit from this field of research such as improvement in hearing-aid [3,4], guiding devices for visually impaired people and navigation system for robots. Although this is a relatively new challenge presented in DCASE 2021 Task 1a, designing low complexity deep learning models has gained traction in the image domain with state-of-the-art models such as MobileNet [5,6], ShuffleNet [7,8], SqueezeNet [9] , CondenseNet [10] and ShiftNet [11]. Another aspect of acoustic scene classification (ASC) framework is selecting the feature extractor, and commonly, log-mel spectrogram is being used to preprocess the raw waveform into a time-frequency representation.

However, in this paper, we distinctly used deep scattering spectrum (DSS) [12,13] which is also a time-frequency representation but unlike log-mel spectrogram, it does not suffer from heavy loss of information when a larger window size (> 23ms) is being applied and further described in Section 2.

Naturally, DSS representation is later fitted with a convolution neural network adapted from MobileNetV2 [5], MobileNetV3 [6] and ShuffleNet[7,8] architecture. Hence, in the subsequence sections, we elaborate on DSS in Section 2, and provide a detailed account of our proposed Mobile Network combined with DSS in Section 3. In Section 4, we described the experimental setup while Section 5 discussed the result. Lastly, we give a conclusion in Section 6.

2. DEEP SCATTERING SPECTRUM

In simplicity, Deep Scattering Spectrum (DSS) is a cascading of wavelet transform and has a very similar computational architecture as CNN [12,13]. 'Morlet' wavelet is the mother wavelet selected for the demodulation of the amplitude and by continuously applying wavelet transform on the demodulated features, which is termed as orders in the context of DSS [13], we retain higher resolution information which was lost during averaging of the earlier orders. Thus, DSS is stable to deformation even when the time scale is larger than 23ms.

Hence, with the understanding of the aforementioned, this paper used DSS as the feature extraction algorithm and DSS is created using Kymatio [14] and constructed with a time scale of ~92ms with a quality factor of 9.

3. MOBILE NETWORK

Mobile networks in this context are dedicated CNN architecture with the goal of reducing computational complexity, such that it has enough depth to classify correctly while being light-weight enough to be ported on device or fast enough for near real-time analysis, and has myriads of application uses, especially for the industry where computational complexity correlate to cost.

Hence, the innovation on mobile networks is not shy from a more monolithic CNN architecture which goal is to increase accuracy. The main algorithm used in mobile network [5-8] [10] to reduce computational complexity is matrix factorization, which changes multiplicative component to additive component. Matrix factorization in CNN is better known as group convolution [7,8] or cardinality [15] and on an extreme level, depthwise convolution layer which is the convolution of a single channel follow by concatenation of all the channels.

Inspired by the works from MobileNet [5,6] and ShuffleNet [7,8], we adapted a reduced convolution layers mobileNetV2 [5], where our model only consists 8 mobile convolution blocks with number of filters as {16,24,24,32,32,64,64,80} and alpha as 0.5, where alpha is a setting which determine the number of channels per convolutional layers in each convolution block [5,6]. Similar to the first convolution layer of [5], we have a convolution layer as the first layer. To reduce the computational cost further, we

Scene Label	*logloss*	*Device-wise log-losses*									*Accuracy*
		A	*B*	*C*	*S1*	*S2*	*S3*	*S4*	*S5*	*S6*	
airport	1.593	1.186	1.696	1.561	1.631	1.451	1.306	1.908	1.653	1.943	46.5%
bus	1.179	0.831	1.373	1.110	1.137	1.116	1.133	1.238	1.252	1.420	79.5%
metro	1.461	1.258	1.421	1.456	1.614	1.519	1.397	1.482	1.466	1.536	58.9%
metro_station	1.507	1.240	1.587	1.571	1.576	1.621	1.521	1.612	1.425	1.408	59.6%
park	1.111	0.728	0.748	0.654	1.167	1.385	1.163	1.330	1.247	1.577	80.8%
public_square	1.880	1.533	1.816	1.575	1.939	1.971	1.884	1.961	1.986	2.260	37.7%
shopping_mall	1.377	1.134	1.309	1.316	1.451	1.315	1.328	1.420	1.295	1.823	70.0%
street_pedestrian	1.737	1.513	1.739	1.580	1.740	1.804	1.714	1.770	1.937	1.838	42.8%
street_traffic	0.818	0.686	0.883	0.810	0.875	0.835	0.788	0.693	0.698	1.099	88.9%
tram	1.434	1.113	1.714	1.368	1.219	1.526	1.393	1.449	1.532	1.591	65.7%
Average	**1.410**	1.122	1.428	1.300	1.435	1.454	1.363	1.486	1.449	1.650	**63.03%**

Table 1. Classification result for DSSMNet1 based on scene wise log loss, device wise log loss and scene wise accuracy. Total Average log loss for our best model is 1.410 while achieving an accuracy of 63.03%.

follow [6] suggestion in adapting a fully convolution approach (FCN), hence, our last two layers consists of a convolution layer with filter size of (1x1x128) follow by another convolution layer with filter size of (1x1x10). Lastly, global average pooling then a softmax is applied as the classifier.

For another model, instead of 8 mobile blocks, we have 6 and our alpha is being set as 1. For the last mobile convolution block, following [8] channel split concept, we split the network into half during the expansion process. We also remove the second last (1x1,128) convolution layer and adapted average channel-wise attention and spatial attention [16]. As having both average and max channel attention will exceed the model complexity criteria, only average channel attention is included.

In brevity, our convolution layers are always coupled with a batch normalization follow by an activation function swish [17], lastly, our models will be termed as DSSMNet1 and DSSMNet2, in respect to the order presented in the earlier paragraphs.

4. EXPERIMENTATION SETUP

In this section, we discussed the dataset being used and provide a description on the new challenge requirement for DCASE 2021 Task1a. Next, we provide details on the training process of our proposed feature and model.

4.1. Dataset

Our model is being tested on DCASE 2021 Task 1a dataset [18] which contains acoustic recording from 10 cities and recorded by 3 real devices and 6 simulated devices giving us a total of 64 hours of audio recording. We follow the given training setup where the training/test split is 70%/30% and no down sampling is being performed when preprocessing the waveform to IDSS.

4.2. Implementation

In this paper, we evaluated on two models, one with DSS as the input representation, while the other uses IDSS as the input representation. Our model is trained in mini batch size of 16 with Rectified Adam [19] with warmup setting that increase from 0 to 0.001 in 1000 steps, then decrease linearly from 0.001 to 0.000001 in 9000 steps. Data Augmentation, Mix Up with alpha of 0.2 is applied only on DSSMNet2 as it yields better performance. The entire CNN system is being built and trained using Tensorflow & Keras and following [2], we further optimized the model using Tensorflow post-training float 16 quantization method. Hence, our models consist of 63448 and 64850 total parameters, which is then converted to float 16, giving us 124 kb and 126.6kb, for DSSMNet1 and DSSMNet2 respectively. Lastly, the classification result is evaluated using our float 16 optimized models.

5. RESULT

Following DCASE 2021 Task1a evaluation metric, Table 1 provide the breakdown of the classification result, for DSSMNet1 model and Table 2 show the comparison of Baseline model vs DSSMNet1 and DSSMNet2. Our best model, DSSMNet1 achieved 63.03% classification accuracy which is ~15% improvement from DCASE 2021 Task1a baseline model, however, if we compare logloss, DSSMNet1 only slightly edge over baseline model while DSSMNet2 achieve a significant decrease of ~0.23. Hence, for DCASE 2021 Task1a challenge, DSSMNet2 stands to be a better model.

6. CONCLUSION

The main challenge here is the limitation of model complexity and model size and hence, it is not possible to relay on transfer learning or embedded system where we can tap on pre-existing models or enriched features to further improve the performance of the system. However, it developed our creativity in adapting various model design to create an optimal model as we need to consider the viability of every layers added to the model. Lastly, the model with the highest accuracy might not always be the optimal model

No	Model	logloss	Accuracy
1	Dcase Task1a baseline [2]	1.473	47.7%
2	DSSMNet1	1.41	**63.03%**
3	DSSMNet2	**1.242**	62.30%

Table 2. Result comparison between our proposed models and Dcase Task1a baseline.

as we need to evaluate with other metric such as log loss and also consider the complexity of the model.

7. REFERENCES

[1] http://dcase.community/challenge2021/.

[2] Irene Martín-Morató, Toni Heittola, Annamaria Mesaros, and Tuomas Virtanen, "Low-complexity acoustic scene classification for multi-device audio: analysis of dcase 2021 challenge systems*",* 2021, arXiv:2105.13734.

[3] Tchorz, Jürgen & Wollermann, Simone & Husstedt, Hendrik, "Classification of Environmental Sounds for Future Hearing Aid Applications" *Elektronische Sprachsignalverarbeitung*, 2017, Volume: 86

[4] A. Hüwel, K. Adiloğlu and J. -H. Bach, "Hearing aid Research Data Set for Acoustic Environment Recognition," *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 706-710, doi: 10.1109/ICASSP40776.2020.9053611.

[5] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov and L. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4510-4520, doi: 10.1109/CVPR.2018.00474.

[6] A. Howard et al., "Searching for MobileNetV3," *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 1314-1324, doi: 10.1109/ICCV.2019.00140.

[7] Xi. Zhang, X. Zhou, M. Lin, and J. Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. *CoRR*, abs/1707.01083, 2017.

[8] Ma, Ningning, X. Zhang, Hai-Tao Zheng and J. Sun. "ShuffleNet V2: Practical Guidelines for Efficient CNN Architecture Design." *ECCV (2018).*

[9] Iandola, F.N., Moskewicz, M., Ashraf, K., Han, S., Dally, W., & Keutzer, K. (2016). SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <1MB model size. ArXiv, abs/1602.07360.

[10] Huang, G., Liu, S., Maaten, L.V., & Weinberger, K.Q. "CondenseNet: An Efficient DenseNet Using Learned Group Convolutions", *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2752-2761.

[11] S Yan Z., Li X., Li M., Zuo W., Shan S. (2018) Shift-Net: Image Inpainting via Deep Feature Rearrangement. In: Ferrari V., Hebert M., Sminchisescu C., Weiss Y. (eds) Computer Vision – ECCV 2018. ECCV 2018. Lecture Notes in Computer Science, vol 11218. Springer, Cham. https://doi.org/10.1007/978-3-030-01264-9_1

[12] Mallat, Stéphane. (2012). Group Invariant Scattering. Communications on Pure and Applied Mathematics. 65. 10.1002/cpa.21413.

[13] andén, Joakim & Mallat, Stéphane. (2013). Deep Scattering Spectrum. IEEE Transactions on Signal Processing. 62. 10.1109/TSP.2014.2326991.

[14] Andreux, Mathieu & Angles, Tomás & Exarchakis, Georgios & Leonarduzzi, Roberto & Rochette, Gaspar & Thiry, Louis & Zarka, John & Mallat, Stéphane & andén, Joakim & Belilovsky, Eugene & Bruna, Joan & Lostanlen, Vincent & Hirn, Matthew & Oyallon, Edouard & Zhang, Sixhin & Cella, Carmine-Emanuele & Eickenberg, Michael. (2018). Kymatio: Scattering Transforms in Python.

[15] Xie, S., Girshick, R.B., Dollár, P., Tu, Z., & He, K. (2017). Aggregated Residual Transformations for Deep Neural Networks. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 5987-5995.

[16] Woo, S., Park, J., Lee, J., & Kweon, I. (2018). CBAM: Convolutional Block Attention Module. *ECCV*.

[17] S Ramachandran, P., Zoph, B., & Le, Q.V. (2017). Swish: a Self-Gated Activation Function. arXiv: Neural and Evolutionary Computing.

[18] Toni Heittola, Annamaria Mesaros, and Tuomas Virtanen, "Acoustic scene classification in dcase 2020 challenge: generalization across devices and low complexity solutions." *In Proceedings of the Detection and Classification of Acoustic Scenes and Events 2020 Workshop (DCASE2020)*, 2020, Submitted. URL: https://arxiv.org/abs/2005.14623.

[19] Liu, L., Jiang, H., He, P., Chen, W., Liu, X., Gao, J., & Han, J. (2020). On the Variance of the Adaptive Learning Rate and Beyond. *ArXiv, abs/1908.03265*.