ACOUSTIC SCENE CLASSIFICATION USING FULLY CONVOLUTIONAL NEURAL NETWORKS AND PER-CHANNEL ENERGY NORMALIZATION

Technical Report

Konstantinos Vilouras

School of Electrical and Computer Engineering
Aristotle University of Thessaloniki, Greece
kvilouras@ece.auth.gr

ABSTRACT

This technical report describes our approach to Task 1 "Acoustic Scene Classification" of the DCASE 2020 challenge. For subtask A, we introduce per-channel energy normalization (PCEN) as an additional preprocessing step along with log-Mel spectrograms. We also propose two residual network architectures utilizing "Shake-Shake" regularization and the "Squeeze-and-Excitation" block, respectively. Our best submission (ensemble of 8 classifiers) outperforms the corresponding baseline system by 16.2% in terms of macro-average accuracy. For subtask B, we mainly focus on a low complexity, fully convolutional neural network architecture, which leads to 5% relative improvement over baseline accuracy.

Index Terms— DCASE 2020, Acoustic Scene Classification, PCEN, Convolutional Neural Network

1. INTRODUCTION

In this year's DCASE challenge, subtask A addresses the problem of classifying recordings from various devices into one of ten predefined classes. In fact, the official development set consists mainly of data from a single recording device A, while a limited number of recordings from secondary devices (real devices B and C, simulated devices s1-s6) is also provided [1]. Audio files are recorded in mono at 44.1 kHz sampling rate and 24-bit resolution. In subtask B, audio is recorded in stereo, 48 kHz/24-bit format with the same device A. Furthermore, data should now be classified into one of three major classes. Since this is a straightforward task, limitation on model size is imposed, thus forcing the entrants to develop low complexity systems.

The rest of the report is organized as follows. Section 2 describes the proposed methods for audio preprocessing as well as data augmentation. Section 3 introduces the architectures used in this task. Results for each subtask are reported in Section 4. At last, conclusion and future work are presented in Section 5.

2. PROPOSED METHODS

2.1. Audio preprocessing

For both subtask A and subtask B, spectrograms are extracted by firstly applying a Hann window of length 2048 samples and 50% overlap to each signal and then computing the Short-time Fourier transform (STFT). Note that all audio files related to subtask B are converted to mono prior to spectrogram extraction. Subsequently, we define the Mel scale using the HTK formula [2] and apply the Mel filter bank to the power spectrum. Minimum frequency is set to 0 Hz, whereas the maximum is set to the Nyquist frequency (22.05 kHz for subtask A, 24 kHz for subtask B). Last, the resulting time-frequency representation is converted to log scale. This type of preprocessing yields spectrograms with 128 frequency bins and 431 (or 469) time samples for subtask A (or subtask B), respectively.

2.2. Per-Channel Energy Normalization (PCEN)

Per-channel energy normalization was introduced in the task of keyword spotting as a technique that improves robustness to loudness variation [3]. Recently, Lostanlen et al. [4] provided insight into how PCEN works through asymptotic analysis. The entire process can be divided into three stages, namely temporal integration, adaptive gain control (AGC) and dynamic range compression (DRC). The first stage, i.e. temporal integration, involves the low-pass filtering of the Mel magnitude spectrum to estimate the level of background noise. This results in a smoothed version of the Mel spectrogram, which is used for adapting the gain level in the following stage. At the final stage, dynamic range compression is applied to adjust the loudness of the foreground regions.

We use the open-source implementation of PCEN provided by librosa[1]. This preprocessing step is only applied to subtask A to compensate for the effect of each recording device on the spectral content of a signal. We adopt the following set of parameters, i.e. time constant $T = 0.06$, the exponent of AGC $\alpha = 0.8$, exponent of DRC $r = 0.25$, while the soft threshold of DRC is empirically set to $\delta = 1000$ to improve the average foreground-to-background ratio. Fig. 1 shows the differences between log-Mel spectrograms and the corresponding PCEN representations for two overlapping audio samples recorded with two different devices.

2.3. Data augmentation

All models are trained using the official train/validation split. For subtask A, instead of using any kind of external data, we decided to apply some data augmentation methods to reduce

[1] https://librosa.github.io/librosa/generated/librosa.core.pcen.html

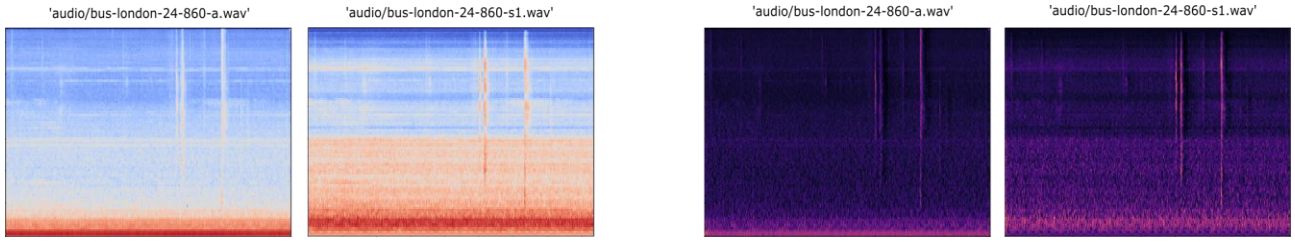

Figure 1: (from left to right) Log-Mel spectrograms vs. PCEN representation for overlapping recordings with real device A and simulated device s1, respectively.

overfitting and improve the generalizability of our models. First, mixup augmentation [5] is used with $\alpha = 0.2$. Additionally, we use the Audiomentations[2] library to apply the following transformations, i.e. frequency masking (minimum bandwidth set to 0.1, maximum set to 0.5), time stretching (minimum rate is 0.8, maximum is 1.2), shifting, and clipping distortion (maximum percentile threshold set to 40). Note that the aforementioned transformations are only applied to data from secondary devices (i.e., all except for recording device A).

3. ARCHITECTURES

Our submission is based exclusively on fully convolutional neural networks. For subtask A, we propose two ResNet variants, while a low complexity network is introduced for subtask B.

Note that each network starts with a batch normalization layer as suggested in [6]. This technique replaces any type of data normalization prior to network input.

3.1. Residual Networks (subtask A)

ResNets [7] are widely used in tasks related to computer vision since they can achieve higher accuracy by increasing the depth of the network. In this task, the first ResNet variant that we propose utilizes "Shake-Shake" regularization [8] which combines two parallel branches stochastically. The residual block for this architecture is depicted in Fig. 2. The input of each block is transformed using 1x1 convolution to match the dimensions of the residual block's output. The final network architecture is detailed in Table 1.

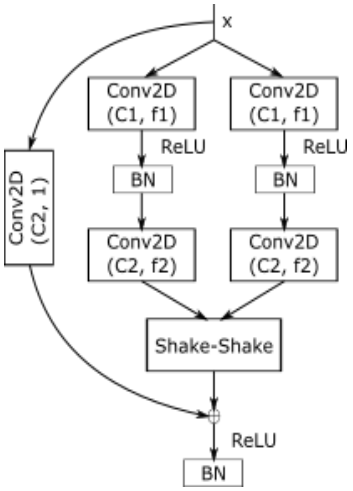

Figure 2: Residual block for ShakeResNet architecture. C1, C2 denote the number of channels, while f1, f2 are the kernel sizes.

[2] https://github.com/iver56/audiomentations

The second ResNet variant is based on the "Squeeze-and-Excitation" (SE) block [9]. The SE block forces the network to adaptively adjust the weighting of each feature map, which leads to more informative features. The structure of the residual block in this case is provided on [9], while Table 2 shows the architecture of this network. We did not employ max pooling layers between successive residual blocks; hence the frequency axis remains unchanged throughout the network. Moreover, we set the reduction ratio to $r = 8$ for the first two residual blocks since the number of channels in these blocks is relatively small. For the last two blocks, the reduction ratio is set to its default value $r = 16$.

3.2. Low complexity network (subtask B)

Due to model size limitation, we adopt a straightforward, VGG-style architecture with an increasing number of filters. We choose ELU as the activation function for every convolutional layer. We also use spatial dropout since it drops entire feature maps instead of individual pixels. Model size reaches 496.2 KB (using 32 bits per parameter), therefore it meets the above constraint. This network architecture is shown in Fig. 3.

3.3. Training

All models are trained using Keras (version 2.3.0) with Tensor-

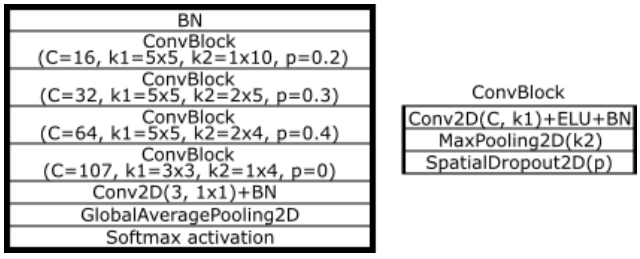

Figure 3: Low complexity, fully convolutional deep neural network architecture. C is the number of channels, k1 and k2 are kernel sizes, whereas p is the dropout rate.

Table 1: ShakeResNet architecture

Layer	Channels	Kernel size
BN	-	-
Conv2D+ReLU+BN	16	5x5
MaxPooling2D	16	1x10
ResidualBlock (C1=24, C2=32)	32	f1 = 5x5 f2 = 5x5
MaxPooling2D	32	2x5
ResidualBlock (C1=48, C2=64)	64	f1 = 5x5 f2 = 5x5
MaxPooling2D	64	2x3
ResidualBlock (C1=96, C2=128)	128	f1 = 3x3 f2 = 3x3
MaxPooling2D	128	1x3
ResidualBlock (C1=192, C2=128)	128	f1 = 3x3 f2 = 1x1
Conv2D+BN	10	1x1
GlobalAvgPooling	10	-
Softmax	-	-

Table 2: SEResNet architecture

Layer	Channels	Kernel size
BN	-	-
Conv2D+ReLU+BN	16	5x5
MaxPooling2D	16	1x10
ResidualBlock (C1=16, C2=24)	24	f1 = 5x5 f2 = 5x5
ResidualBlock (C1=32, C2=48)	48	f1 = 5x5 f2 = 5x5
ResidualBlock (C1=64, C2=96)	96	f1 = 3x3 f2 = 3x3
ResidualBlock (C1=128, C2=192)	192	f1 = 3x3 f2 = 3x3
Conv2D+BN	10	1x1
GlobalAvgPooling	10	-
Softmax	-	-

flow (version 2.1.0) as backend. We use the Adam optimizer [10] with an initial learning rate of 0.0005 and the cross-entropy loss function. The learning rate is decreased by a factor of 0.1 (lower bound is equal to 0.0001) if the validation loss does not improve for 5 consecutive epochs. We train each model for 100 epochs with a batch size of 16 and data shuffling between epochs. During training, we save the best performing model based on validation accuracy. Moreover, all network weights are initialized using the He normal technique. L2 regularization is also added on all convolutional layers with different values of lambda for each network. In the following paragraphs we detail the training setup for each subtask separately. For evaluation, we trained all models on the entire development set.

For subtask A, each model is trained on the provided training set independently using either log-Mel spectrograms or PCEN representations. This process results in 4 distinct models. Additionally, we train 4 models using the augmented dataset. Note that lambda is set to 0.0001 for all models related to this subtask.

For subtask B, we train a single model using exclusively log-Mel spectrograms. Here, lambda is set to 0.001, which leads to a highly regularized model.

3.4. Ensemble models

Our submission for subtask A consists only of ensembles which are created by averaging each model's softmax predictions. As a result, we combine several weak learners, which are complementary to each other, to form a robust model (strong learner). This method significantly boosts the overall prediction performance. Our final systems are labeled as follows.

- **Vilouras_AUTh_task1a_1**: Ensemble of 4 models trained on the official development set (each residual network is trained on either log-Mel spectrograms or PCEN representations, respectively).
- **Vilouras_AUTh_task1a_2**: Ensemble of 4 models trained on the augmented dataset (using either log-Mel spectrograms or PCEN representations).
- **Vilouras_AUTh_task1a_3**: Ensemble of 8 models, fusion of the previous two ensembles.

4. RESULTS

In this section we present the results on the official validation set for each subtask. Macro-average accuracy and log loss are used as evaluation metrics for this task. Table 3 illustrates the class-wise and the average accuracy, whereas Table 4 shows the per-class log loss as well as the average over each class, respectively. Each column represents the corresponding ensemble as labeled in the previous section (e.g. Sub. 1 refers to the ensemble with submission ID equal to 1). Finally, Table 5 displays the overall results associated to subtask B.

5. CONCLUSION

In this technical report, we elaborate our approach for Task 1 of the DCASE 2020 challenge. We propose various deep convolutional neural network architectures which, combined with data augmentation and regularization, reach state-of-the-art results. In subtask A, we introduce per-channel energy normalization as a novel preprocessing step to address data mismatch, i.e. audio samples recorded with multiple devices. Furthermore, we employ deep residual networks for classification. Each model is trained on either PCEN representations or log-scaled Mel spectrograms. Since the resulting classifiers complement each other, we decided to submit 3 ensembles. Our best performing ensemble outperforms the corresponding baseline system by 16.2% in terms of macro-average accuracy. In subtask B, we implement a low complexity system trained exclusively on log-Mel spectrograms. This model surpasses the baseline by 5%.

Future work will focus on further optimizing PCEN hyperparameters. Although values of α close to 1 can adequately eliminate spectral equalization [4], training on these representations proves to be extremely unstable and inefficient. Finally, we

Table 3: Subtask A, class-wise and average accuracy

Class	Sub. 1	Sub. 2	Sub. 3
Airport	64.7 %	56.2 %	61.3 %
Bus	85.2 %	87.2 %	88.9 %
Metro	52.2 %	65.0 %	60.9 %
Metro station	60.9 %	64.7 %	65.7 %
Park	91.9 %	90.2 %	92.6 %
Public square	54.6 %	50.2 %	52.9 %
Shopping mall	71.7 %	74.4 %	75.4 %
Street pedestrian	43.8 %	52.2 %	49.5 %
Street traffic	81.1 %	85.2 %	84.2 %
Tram	75.1 %	66.3 %	71.7 %
Average	**68.1 %**	**69.2 %**	**70.3 %**

Table 4: Subtask A, per-class and average log loss

Class	Sub. 1	Sub. 2	Sub. 3
Airport	0.977	1.146	1.031
Bus	0.500	0.499	0.477
Metro	1.122	0.912	0.983
Metro station	1.115	0.998	1.022
Park	0.394	0.436	0.405
Public square	1.380	1.471	1.389
Shopping mall	0.756	0.759	0.739
Street pedestrian	1.347	1.228	1.259
Street traffic	0.651	0.514	0.567
Tram	0.839	0.937	0.851
Average	**0.908**	**0.890**	**0.872**

Table 5: Results of subtask B

Class	Accuracy	Log loss
Indoor	89.6 %	0.281
Outdoor	89.3 %	0.256
Transportation	97.9 %	0.083
Average	**92.3 %**	**0.211**

plan to investigate class-conditional data augmentation to improve the performance of our proposed models.

6. ACKNOWLEDGMENT

The author would like to thank Dr. Charalampos Dimoulas, Lazaros Vrysis and Iordanis Thoidis for their supervision and guidance.

7. REFERENCES

[1] 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

[2] S. J. Young, D. Kershaw, J. Odell, D. Ollason, V. Valtchev, and P. Woodland, *The HTK Book Version 3.4*. Cambridge University Press, 2006.

[3] Y. Wang, P. Getreuer, T. Hughes, R. F. Lyon, and R. A. Saurous, "Trainable frontend for robust and far-field keyword spotting," in *Proc. IEEE ICASSP*, 2017.

[4] V. Lostanlen *et al*., "Per-Channel Energy Normalization: Why and How," in *IEEE Signal Processing Letters*, vol. 26, no. 1, pp. 39-43, Jan. 2019, doi: 10.1109/LSP.2018.2878620.

[5] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "Mixup: Beyond empirical risk minimization," in *International Conference on Learning Representations*, 2018.

[6] M. D. McDonnell and W. Gao, "Acoustic Scene Classification Using Deep Residual Networks with Late Fusion of Separated High and Low Frequency Paths," *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Barcelona, Spain, 2020, pp. 141-145, doi: 10.1109/ICASSP40776.2020.9053274.

[7] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770– 778.

[8] X. Gastaldi, "Shake-shake regularization," *arXiv preprint arXiv:1705.07485*, 2017.

[9] J. Hu, L. Shen, and G. Sun, "Squeeze-and-Excitation Networks," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, 2018, pp. 7132-7141, doi: 10.1109/CVPR.2018.00745.

[10] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. [Online]. Available: http://arxiv.org/abs/1412.6980