BUET BOSCH CONSORTIUM (B2C) ACOUSTIC SCENE CLASSIFICATION SYSTEMS FOR DCASE 2017 CHALLENGE

Rakib Hyder[1]*, Shabnam Ghaffarzadegan*[2]*, Zhe Feng*[2]*, Taufiq Hasan*[1]

[1]Bangladesh University of Engineering and Technology (BUET), Dhaka, Bangladesh.
[2]Robert Bosch Research and Technology Center (RTC), Palo Alto, CA.

rakib.hyder.bd@ieee.org, taufiq@bme.buet.ac.bd
{shabnam.ghaffarzadegan,Zhe.Feng2}@us.bosch.com

ABSTRACT

This technical report describes the systems jointly submitted by Bangladesh University of Engineering and Technology (BUET), Dhaka, Bangladesh, and Robert Bosch Research and Technology Center, Palo Alto, CA, USA, for the Acoustic scene classification (ASC) task of the DCASE 2017 challenge. Our sub-systems mainly consist of Convolutional Neural Network (CNN) based models trained on Spectrogram Image Features (SIF) using Mel and Log-scaled filter-banks. We also used a novel multi-band approach that learns the CNN models from different frequency bands separately using a single spectrogram. In a variant of CNN sub-systems, large dimensional audio segment level feature vectors, termed as super vectors (SV), are extracted from the flattening layer of a trained CNN model. These features are later classified utilizing a Probabilistic Linear Discriminant Analysis (PLDA) model. This sub-system is termed as the CNN super vector (CNN-SV) system [1]. We also implemented an MFCC feature based GMM super vector (GSV) system with a PLDA classifier, and an acoustic feature ensemble based feed-forward Neural Network (NN) system. Finally, we utilized linear score-fusion to combine the class-wise scores obtained from the different sub-systems.

***Index Terms*—** Acoustic scene classification, convolutional neural networks, super vector, score fusion.

1. INTRODUCTION

In this report, we describe four systems proposed for Task 1 (ASC) in the DCASE-2017 challenge [2]. We provide the performances of our systems on the development dataset. At first, we train a Deep Convolutional Neural Network (DCNN) using spectrograms of audio excerpts. This constitutes our baseline CNN system. In a later stage, we extracted the flattening layer output of the DCNN to form super vectors (SV). After some post-processing, these SVs are fed to a PLDA classifier to form our CNN-SV-PLDA system. We used Mel-scaled and Log-scaled spectrograms as features to train our CNN systems and subsequently obtain the CNN-SVs.

We also propose a new architecture for CNN based acoustic scene classification, termed multi-band CNN. In this framework, we divide a spectrogram into different regions in the frequency dimension and trained each segment independently. Later, we merged the outputs of the CNNs in the flattening layer. It is also possible to extract SVs using this approach and use the PLDA classifier as in the CNN-SV-PLDA systems [1].

Other than CNN-SV-PLDA and multi-band CNN-SV-PLDA systems, we also used a Gaussian Mixture Model (GMM) and single layer Neural Network (NN) based system. In the GMM based system, adapted GMM mean SVs are extracted from each audio segment which are then classified using a PLDA model. The GMM-SV-PLDA system does not perform on par with the CNN-SV systems. However, it improves the accuracy of the overall system during fusion. Finally, we used a feed-forward Neural Network (NN) based system trained with different functionals of spectral, prosodic and acoustic features.

We utilized various combination of the above sub-systems and fused their scores to prepare the submissions for this task. We utilized the FoCal toolkit [3] for logistic regression based fusion parameter learning.

2. DATASET

In this work, we utilize the DCASE 2017 acoustic scene classification challenge data [2]. The dataset consists of audio samples from 15 (fifteen) different indoor and outdoor locations or environments. There are 4680 and 1620 audio segments in the development and evaluation data, respectively. The 2-channel audio segments are 10s in duration and are recorded in a 24-bit PCM format at $44,100$Hz sampling rate. The ASC development dataset is designed as a four-fold cross-validation task with about 75% data used for training and the remaining 25% for testing. The average accuracy over the folds is used as the performance evaluation metric [2].

3. GMM SUPER VECTOR (GSV) SYSTEM

The GSV framework was first utilized in speaker recognition [4]. This method generates a high-dimensional vector by concatenating the GMM mean vectors that model specific audio segments using the Maximum *A Posteriori* (MAP) adaptation [5]. These SVs are then used as features for PLDA classifier.

3.1. Features

For the GMM-SV system, we extract 60 dimensional Mel-frequency cepstral coefficients (MFCC) [6], where 19 static coefficients are computed including C_0, and the velocity (Δ) and acceleration ($\Delta + \Delta$) coefficients are appended.

3.2. GMM adaptation and super vector extraction

Initially, a GMM is trained on the DCASE 2017 training data for the corresponding fold. This is a generic acoustic scene independent model, known as the Universal Background Model (UBM) in

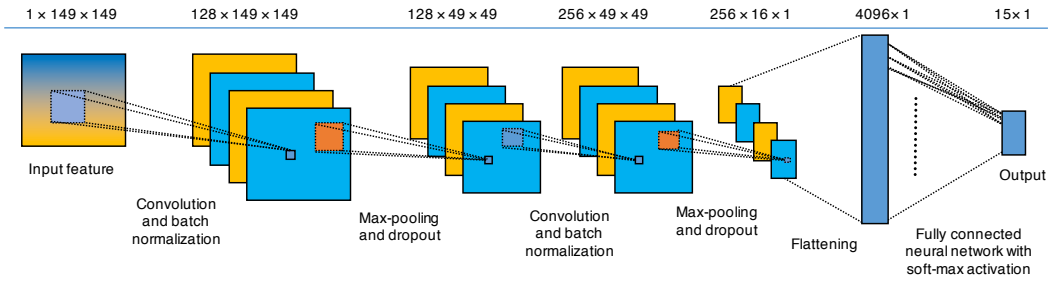

Figure 1: A flow-diagram of the utilized CNN architecture for the experiments using spectrogram image features. The proposed CNN SVs are formed from the flattening layer activations which are input to softmax output layer.

speaker recognition literature [5,7]. Next, audio segment dependent GMM parameters are estimated using MAP adaptation. The GMM SVs are obtained by concatenating the adapted mean vectors as of each audio segment. We use 64 component GMMs trained using an Expectation Maximization (EM) algorithm [8] with 5 iterations per mixture split. For MAP adaptation, a relevance factor of 14 is used. Finally, 3840 (64×60) dimensional SVs are extracted from each training and test segment. Further details of this system can be found in [1].

3.3. SV post-processing

We first perform mean normalization across the training SVs. Next, we divide each vector by it's own L^2 norm for length normalization [9]. The resulting vectors are then reduced in dimension to 14 using a Linear Discriminant Analysis (LDA) projection and normalized using the Within Class Covariance Normalization (WCCN) [10]. The parameters required in the post-processing steps are learned from the training data only and applied on the evaluation data.

3.4. Probabilistic Linear Discriminant Analysis (PLDA)

We utilize a Gaussian PLDA classifier with a full-covariance residual noise [9]. In this model, an R dimensional post-processed SV extracted from audio segment s is expressed as:

$$\mathbf{m}_s = \mathbf{m}_0 + \mathbf{\Phi}\beta + \mathbf{n}. \quad (1)$$

Here, $\mathbf{m}_0 \in \mathbb{R}^R$ is the acoustic scene independent mean vector, $\mathbf{\Phi}$ is an $R \times N$ low rank matrix representing the scene dependent basis functions, $\beta \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ is an $K \times 1$ hidden vector, and $\mathbf{n} \in \mathbf{R}^R$ is a random vector representing the full covariance residual noise. We train our model using the averaged post-processed SV obtained from each class and set $K = 14$. The training data of the corresponding fold was used to train the PLDA model. The scoring is performed as described in [11]. To determine if $\mathbf{m}_i$ and $\mathbf{m}_j$ belong to the same class (H_1) or not (H_0), we use the following likelihood ratio:

$$\mathcal{L}_{i,j} = \frac{P(\mathbf{m}_i, \mathbf{m}_j | H_1)}{P(\mathbf{m}_i | H_0) P(\mathbf{m}_j | H_0)}. \quad (2)$$

This comparison is performed across all training and test segments to determine the highest scoring class for each test.

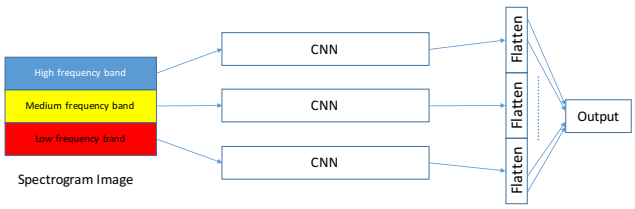

Figure 2: A flow-diagram of the utilized Multi-band CNN architecture for the experiments using spectrogram image features. Multi-band CNN SVs are formed from the flattening layer.

4. CNN SYSTEMS

4.1. Spectrogram Image Features (SIF)

We use a single channel spectrogram excerpt with a dimension of 149×149 as input to our CNN system. This setup closely follows the top-scoring system in DCASE 2016 [12]. First, the audio data is down-sampled to a rate of 22,050Hz and segmented at 31.25 frames/sec. Next, Short-Time Fourier Transform (STFT) is computed on 2048 sample time windows. In [12], a logarithmic filterbank was used with 24 bands per octave extracted within a passband of 20Hz to 11.025kHz with the MADMOM toolkit [13] which resulted in 149 frequency bins. Then the spectrogram was segmented into 149×149 frames with 25% overlap in the temporal dimension. These spectrogram excerpts are the inputs to the CNN system for an audio segment. We also used Mel-scaled filterbanks with the same number of frequency bins to generate corresponding SIF segments from each STFT window.

4.2. Baseline CNN architecture

We followed [14] with some modifications to build our deep CNN architecture. A block diagram representation of our model is presented in Fig. 1. The first layer performs a convolution over the input spectrogram with 128 kernels with 3×3 kernel size and unitary depth and stride in both dimensions. The obtained feature maps are then sub-sampled using a max-pooling layer operating over 3×3 non-overlapping squares. The second convolutional layer is very similar to the first one except with a higher number of kernels (256 instead of 128). The second and last sub-sampling operation is performed aiming to remove the temporal axis. Therefore, we use a max-pooling layer which operates over the entire sequence length and, on the frequency axis, only over 3 non-overlapping frequency bands. Rectified linear unit (ReLU) [15] activation functions are used for the kernels in both convolutional layers. Finally, to classify the audio segment in 15 classes, the output layer consists of a

ID	Feature	Classifier	% Accuracy				
			Fold1	Fold2	Fold3	Fold4	Average
Sys. 1	60D MFCC+Δ+ΔΔ	GMM-SV-PLDA	81.11	80.22	81.02	81.97	81.08
Sys. 2	149×149 Log-scaled SIF	Baseline CNN	81.28	81.93	79.78	80.42	80.85
Sys. 3	149×149 Log-scaled SIF	CNN-SV-PLDA	84.27	81.15	83.43	84.36	83.30
Sys. 4	149×149 Mel-scaled SIF	Baseline CNN	79.91	81.10	79.04	83.93	81.00
Sys. 5	149×149 Mel-scaled SIF	CNN-SV-PLDA	82.65	81.25	83.92	84.70	83.13
Sys. 6	149×149 Log-scaled SIF	Multi-band CNN	83.85	80.81	80.02	83.42	82.02
Sys. 7	149×149 Log-scaled SIF	Multi-band CNN-SV-PLDA	**85.47**	**84.23**	84.28	84.02	**84.50**
Sys. 8	149×149 Mel-scaled SIF	Multi-band CNN	82.31	80.31	77.23	83.93	80.95
Sys. 9	149×149 Mel-scaled SIF	Multi-band CNN-SV-PLDA	83.33	83.90	**84.43**	**85.21**	84.22
Sys. 10	272D Acoustic Feature set	NN	76.06	72.19	73.62	74.01	73.97

Table 1: Performance evaluation of the CNN, GMM-SV-PLDA and CNN-SV-PLDA systems with different spectrogram features on the DCASE 2017 ASC development dataset. %Accuracy for each fold and their average values are reported.

ID	System	%Accuracy (Dev)	%Accuracy (Eval)
Submission-1	Linear equal-weight score fusion of Systems (1,3,5,7,9)	88.7	**74.1**
Submission-2	Log-scaled SIF CNN-SV-PLDA	83.3	72.2
Submission-3	Linear logistic regression fusion of Systems (1,2,4,6,7,8,9)	**89.8**	68.6
Submission-4	Linear logistic regression fusion of Systems (1–10)	89.6	72.0

Table 2: The constituents of the systems submitted to DCASE 2017 ASC challenge. The accuracy is obtained by averaging the accuracies from the four folds of the DCASE 2017 training data.

15-node fully-connected neural network with a softmax activation function.

4.3. Regularization, Optimization and Model Training

We utilized batch normalization [16] as an intermediate layer after each of the two convolutional layers. We also used a dropout layer for regularization (dropout $= 0.25$) after each of the two max-pooling layers. The CNN system is implemented with the Keras Python library. We utilized the categorical cross-entropy loss function and the Adaptive Momentum (ADAM) [17] optimization approach. We set exponential decay rate for the moment estimates of the ADAM algorithm as $\beta_2 = 0.99999$ and $\beta_1 = 0.9$ to reduce the weights on the previous time stamps. We kept the default value for the parameter, ϵ (10^{-8}). Finally, we used a learning rate of 0.0001 with 200 training epochs. We observed the loss in training dataset for every epoch and used the model with least loss. The classification decision for an audio segment is calculated by averaging the prediction scores obtained from the short segments.

4.4. The CNN super vector (CNN-SV) system

In order to combine the feature learning strength of CNN with the SV back-end, we formed a high dimensional vector concatenating the activations from the flattening layer (Fig. 1) of a trained CNN system and fed it as an input to the PLDA back-end. In this system, training is performed in two stages. In the first stage, the CNN model is trained on the SIF features on the training dataset as described in Sec. 3. Once the model is trained, all the training and test data is evaluated using the CNN model and the flattening layer activations are combined to form a high-dimensional super vector (SV) similar to Sec. 3 (e.g. 4096 dimensions in case of log-scaled SIF). Next, the extracted training CNN-SV features are post-processed by LDA, WCCN and length normalization according to Sec. 3.3. The processed CNN-SV features are used to train the PLDA model as described in Sec. 3.4.

4.5. Multi-band CNN System

In the multi-band CNN approach, the input spectrogram is separated into different frequency regions and each of these regions are provided for training separate CNN models. A block diagram explaining the multi-band CNN system is shown in Fig. 2. In our system we segmented the spectrogram with 149 frequency bins into 5 overlapping segments with an overlap of 10 frequency bins. We applied the same CNN system described above for each segments. We also applied the dense layer with same activation function (softmax) on the merged flattening layer. The loss function, optimizers and regularizers also remain the same.

5. NEURAL NETWORK (NN) BASED SYSTEM

5.1. Acoustic feature set

We used a large dimensional acoustic feature set by calculating functionals of various spectral and prosodic features. Different features include: 13-dimensional MFCC, delta of MFCC, zero crossing rate, delta of zero crossing rate, RMS, delta of RMS, spectral centroid, delta of centroid, pitch and delta of pitch. The functionals used are: minimum, maximum, median, mean, standard deviation, skewness and kurtosis. Using these functionals from the above features, we obtain the 272 dimensional feature set.

5.2. NN classifier

This system consist of a feed-forward NN with 500 nodes. Rectified linear units (ReLU) are used as activation functions. The output layer consists of 15 nodes with sofmax activation functions. We used the same optimizer and loss function as the CNN system described above.

6. RESULTS

The results of the developed sub-systems are summarized in Table 1. There are in total 10 sub-systems with unique system IDs provided in the first column. From the results, we observe that the CNN based systems perform better than the other systems. Particularly, CNN-SV-PLDA and multi-band CNN-SV-PLDA systems providing superior performance. The multi-band CNN-SV-PLDA trained on log-scaled SIF provides the best accuracy of 85.5% averaged over all folds.

7. FUSION SYSTEMS

To further improve the system performance, we performed linear score fusion of various combinations of the 10 sub-systems. The fusion process consists of first mean and range normalization (divide by the absolute maximum value) of scores obtained from the 15 classes in each test segment. The normalized scores are then averaged across different systems to obtain the fused scores. For our first submission, we used an equal weight linear fusion. For our submissions 3 and 4, we utilize the FoCal toolkit [3] for learning the fusion weights of different systems. We used a logistic regression based fusion algorithm provided in the toolkit. The weights learned from the four folds using the labels of the test data is averaged to learn the final fusion weights. These final weights are then applied to the scores obtained on the evaluation data. For submission 3, we select the best 7 systems to perform fusion. These systems are selected by attempting to fuse all combinations of n ($1 < n \leq 10$) systems selected from the 10 systems and fusing them using the FoCal toolkit. Using this brute-force method, we found that the selected systems in submission 3 provide the best averaged accuracy on the DCASE 2017 development dataset. For submission 4 we simply fuse all 10 systems. The submissions are summarized in Table 2 with the performance obtained on the DCASE 2017 development and evaluation data (averaged across all folds). From the final published results, we observe that Submission-1 has performed the best among our submissions with an accuracy of 74.1%. In DCASE 2017, our team ranked 4th among all teams and our top submission ranked 8th among all the submissions.

8. CONCLUSIONS

In this report, we have described the Acoustic Scene Classification (ASC) systems submitted to the DCASE 2017 challenge by B2C (BUET BOSCH Consortium). The systems included individual CNN based systems trained on Mel and Log-scaled SIF, CNN-SV systems with a PLDA classifier, a GSV system with a PLDA classifier, and a feed-forward NN system with an ensemble of acoustic features. The submissions prepared included individual and fusion systems trained using a logistic regression method.

9. REFERENCES

[1] R. Hyder, S. Ghaffarzadegan, Z. Feng, J. H. Hansen, and T. Hasan, "Acoustic scene classification using a CNN-supervector system trained with auditory and spectrogram image features," in *Proc. InterSpeech*, Stockholm, Sweden, Aug. 2017.

[2] "Detection and Classification of Acoustic Scenes and Events 2017," http://www.cs.tut.fi/sgn/arg/dcase2017/.

[3] N. Brummer, "Focal multiclass toolkit," *URL: http://niko. brummer. googlepages. com/focalmulticlass*, 2014.

[4] W. Campbell, D. Sturim, D. Reynolds, and A. Solomonoff, "SVM based speaker verification using a GMM supervector kernel and NAP variability compensation," in *Proc. IEEE ICASSP*, May 2006, pp. 97–100.

[5] D. Reynolds, T. Quatieri, and R. Dunn, "Speaker verification using adapted Gaussian mixture models," *Digital Signal Process.*, vol. 10, no. 1–3, pp. 19 – 41, 2000.

[6] S. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Trans. Acoustics, Speech, Signal Processing*, vol. 28, no. 4, pp. 357–366, Aug 1980.

[7] T. Hasan and J. H. L. Hansen, "A study on universal background model training in speaker verification," *IEEE Trans. Audio, Speech, Lang. Process.*, pp. 1890–1899, Sep. 2011.

[8] T. K. Moon, "The expectation-maximization algorithm," *IEEE Signal Processing Magazine*, vol. 13, no. 6, pp. 47–60, 1996.

[9] D. Garcia-Romero and C. Y. Espy-Wilson, "Analysis of i-vector length normalization in speaker recognition systems," in *Proc. Interspeech*, Florence, Italy, Oct. 2011, pp. 249 – 252.

[10] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front end factor analysis for speaker verification," *IEEE Trans. Audio. Speech. Lang. Processing*, vol. 19, no. 4, pp. 788–798, 2011.

[11] P. Kenny, "Bayesian speaker verification with heavy tailed priors," in *Proc. Odyssey*, Brno, Czech Republic, 2010.

[12] H. Eghbal-Zadeh, B. Lehner, M. Dorfer, and G. Widmer, "CP-JKU submissions for DCASE 2016: A hybrid approach using binaural i-vectors and deep convolutional neural networks," in *Workshop on DCASE 2016*, Budapest, Hungary, Sep. 2016.

[13] S. Böck, F. Korzeniowski, J. Schlüter, F. Krebs, and G. Widmer, "madmom: A new Python Audio and Music Signal Processing Library," in *Proc. ACM Mult. Conf.*, Amsterdam, The Netherlands, 10 2016, pp. 1174–1178.

[14] M. Valenti, A. Diment, G. Parascandolo, S. Squartini, and T. Virtanen, "DCASE 2016 acoustic scene classification using convolutional neural networks," in *Workshop on DCASE 2016*, Budapest, Hungary, Sep. 2016.

[15] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks." in *Aistats*, vol. 15, no. 106, 2011, p. 275.

[16] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," *arXiv:1502.03167*, pp. 1–11, 2015.

[17] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.