GENERATIVE ADVERSARIAL NETWORK BASED ACOUSTIC SCENE TRAINING SET AUGMENTATION AND SELECTION USING SVM HYPER-PLANE

Seongkyu Mun
Korea University
Dept. of Visual Information Processing, Seoul, 136-713, South Korea
skmoon@ispl.korea.ac.kr

Sangwook Park
Korea University
School of Electrical Eng., Seoul, 136-713, South Korea
swpark@ispl.korea.ac.kr

David K. Han
Office of Naval Research, Arlington VA, USA
ctmkhan@gmail.com

Hanseok Ko
Korea University
School of Electrical Eng., Seoul, South Korea
hsko@korea.ac.kr

ABSTRACT

Although it is typically expected that using a large amount of labeled training data would lead to improve performance in deep learning, it is generally difficult to obtain such DataBase (DB). In competitions such as the Detection and Classification of Acoustic Scenes and Events (DCASE) challenge Task 1, participants are constrained to use a relatively small DB as a rule, which is similar to the aforementioned issue. To improve Acoustic Scene Classification (ASC) performance without employing additional DB, this paper proposes to use Generative Adversarial Networks (GAN) based method for generating additional training DB. Since it is not clear whether every sample generated by GAN would have equal impact in classification performance, this paper proposes to use Support Vector Machine (SVM) hyper plane for each class as reference for selecting samples, which have class discriminative information. Based on the cross-validated experiments on development DB, the usage of the generated features could improve ASC performance.

Index Terms— acoustic scene classification, generative adversarial networks, support vector machine, data augmentation, decision hyper-plane

1. INTRODUCTION

One of the fundamental issues in deep learning is availability of large labeled data set. It has been consistently shown over the last decade that larger labeled data set with deeper network layers can lead to improved results. However, it is not easy to collect large amounts of labeled data, so it is necessary to extract the maximum performance with a small amount of data depending on the application. An example of such constraint is the case of the IEEE DCASE challenge Task 1 for ASC [1-3]. Although it has been well known that the given ASC DB of the competition is insufficient for high classifier performance, there has not been much attempt on augmenting the insufficient amount of data among the participants by using methods such as semi-supervised learning (or pre-training) employing additional databases [4-7]. This is because one of the rules in DCASE challenge prohibits the use of external DBs other than the DCASE Task 1 development set. Obviously, pre-training network using additional DB larger than the development set could improve ASC performance, as shown in our previous research [8]. However, this is not allowed in the DCASE challenge.

Therefore, to improve ASC performance without employing additional DB, our DCASE 2017 work focuses on DB generation. To generate new samples using only the development DB, we propose to use GAN models. The GAN learns two sub-networks: a generator and a discriminator. The discriminator reveals whether a sample is generated or real, while the generator produces samples to pass through the discriminator as real data. The GANs are first proposed by Goodfellow et al. [7] to generate images and gain insights into neural networks. Then, Deep Convolutional GANs (DCGANs) [9] addressed the issue of instability inherent in training GAN. The discriminator of DCGAN can serve as a robust feature extractor. On the other hand, GANs also demonstrate potential in generating images for specific applications. Pathak et al. [10] proposed an encoder-decoder method for image inpainting, where GANs are used as the image generator. Several researches have attempted to use the GAN generated samples as training samples. For labeling the generated samples, the generated samples were all taken as one class in the discriminator in [4-5]. Zheng [6] adopted a novel regularization approach by assigning a uniform label distribution to the generated samples. Although additional data generated by GAN may lead to improved classifier training, it is not clear whether every data point generated by GAN would have equal impact in classifier performance. As it has been shown by SVM, those support vectors that reside near decision boundary are generally crucial in providing key information in classification [16]. We believe that performance could be improved by selecting the generated data by measuring decision value (distance) from decision hyper-plane of SVM for each class.

Recently, GAN has been applied to several acoustic applications, such as voice conversion, speech synthesis and speech enhancement [11-13]. These applications have reference signals for training, such as the same contents of speech set that multiple speakers uttered [11], reference speech already generated by conventional synthesis methods [12], or noisy/clean speech sample pairs [13]. In the case of classifications, typically there is no reference signal which the generator can be built up from. Therefore, instead of training the GAN based speech sample (raw waveform) generator, we propose to use the GAN as an ASC feature generator.

For ASC feature extraction, we used a combined structure of LSTM and CNN with inputs, such as spectrogram and log Mel-Filter Bank (MFB) energy. Using the extracted ASC features, a SVM hyper-plane and a GAN generator for each class were trained. Using the GAN generator, sample pool for each class were generated. Afterwards, based on the criterion of SVM deci-

sion value and the classification rate on the seen/unseen validation DB, feature sampling and new SVM training are conducted iteratively. We used the feature set configuration, which shows the highest performance on seen and unseen data, as the final training DB. More details will be covered in Section 2.2.

2. PROPOSED FRAMEWORK

The process of the proposed GAN based framework is depicted in Figure 1. Following the development DB setup of the baseline system [14], we divided the development DB at 3: 1 ratio for training and validation. For validating the GAN generated samples, we divided the training part in half, Tr-A and Tr-B in Figure 1. The GAN based feature generation and selection were done individually for each class. Therefore, a total of 15 GANs were trained. After the feature samples were generated and selected by GAN and SVM, the augmented feature sets were used for training and validation with Fully Connected Neural Network (FCNN) and SVM for final classification. For improving performance, we conducted late fusion on SVM and FCNN results.

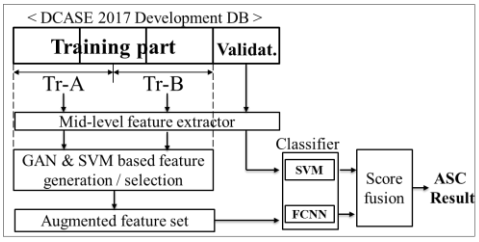

Figure 1: Block-diagram of the proposed framework

2.1. Mid-level ASC feature extraction

For spectrogram or Mel filter-based feature input, various types of networks have been studied in DCASE 2016 [3]. Among the various approaches, we chose a structure of parallel combination of LSTM and CNN [1] to extract both sequential and local time-frequency associated information. Using the network in Figure 2, we could get the classification results directly for the development DB, but we extracted the mid-layer values of network as ASC feature for further processing. For more information on mid-level feature extraction, see [15] or [8] for the visual object classification or ASC. As mentioned in the introduction, due to difficulty of generating raw waveforms using GANs, we used the mid-layer values as ASC features to train GAN and generate 'fake' data.

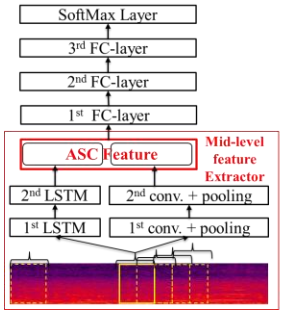

Figure 2: The neural net structure for the feature extractor

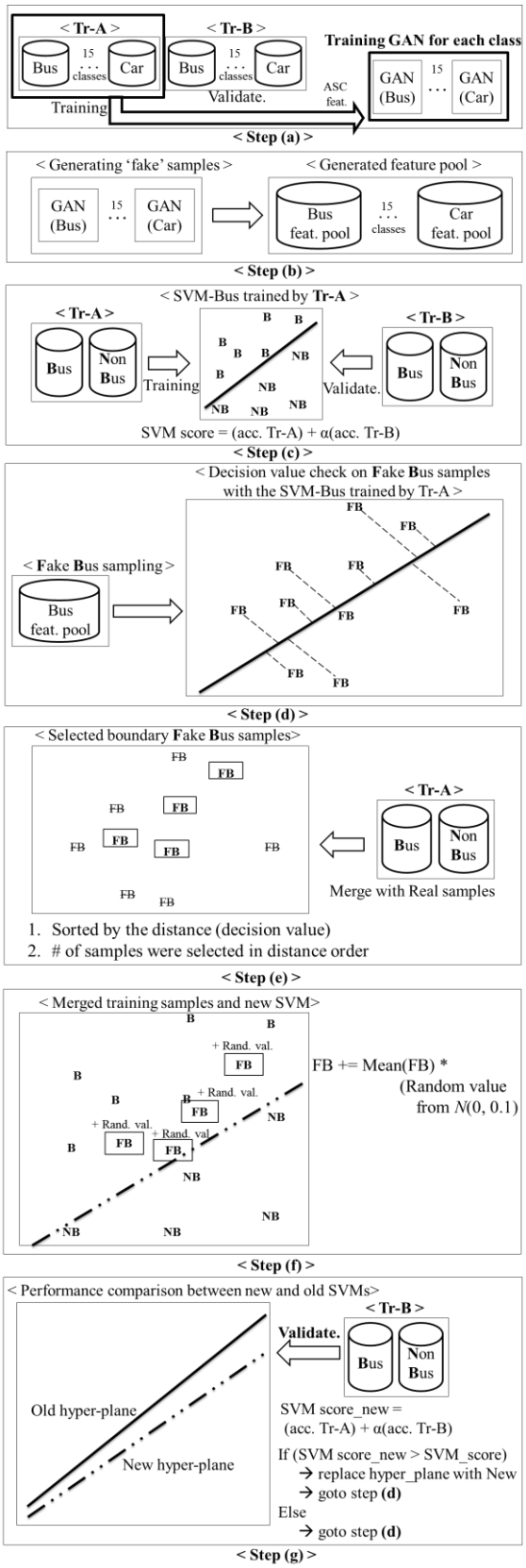

Figure 3: The iterative routine of the DB generation/selection

2.2. Generative adversarial net based training feature set augmentation

The process of the proposed GAN based feature generation and selection are depicted in Figure 3. As shown in step (a), a GAN for each class was trained using the part of the development set, which excludes the validation part for following steps. More details of GAN configuration will be covered in Section 3.2.

Using the trained GANs, we generated 'fake' samples and organized the sample feature pools for each class as shown in step (b). Before using the generated samples, an SVM hyper-plane for each class (target class vs. the others) was first determined from the real data set to establish a baseline performance. We chose the bus class as an example. Note that half of the training set was used for training and the other half was used for validating SVM performance. As shown in step (c), we checked classification performance of SVM with the sum of the training and validation set accuracy. Considering the SVM update in the next step, we added a weight (α, which is bigger than 1) to the unseen data, i.e. validation accuracy.

In step (d), we subsampled 'fake bus' features from the generated bus feature pool and checked decision values on the SVM hyper-plane trained from Tr-A set. As shown in step (e), we sorted the fake samples by the distance order, and chose a preset number of the nearest samples. Additionally, we also included small number of samples near the hyper-plane that were classified as non-bus by handicapping their decision value. We then merged the near boundary fake samples with the real samples of Tr-A set. Step (f) shows the new SVM hyper-plane trained by the merged set. Before training the new SVM, we added random vectors, which are scaled to the magnitude of the samples, to reduce the sample bias of the generation using GAN. As was done in step (c), the classification performance of new SVM was checked with the sum of the training (Tr-A) and validation set (Tr-B) accuracy. If the accuracy score of the new SVM outperforms the previous SVM score, the reference SVM hyper-plane was replaced with the new one and the iteration continues again with the fake sample subsampling in the step (d). If not, the iteration proceeds to the step (d) without replacing the reference hyper-plane. All steps of subsampling, sorting, selecting, merging and performance checking are repeated until the iterative process reaches a preset number of rounds or the performance converged. Once the SVM performance is optimized, the associated support vectors of fake bus features were used for the augmented training set. The entire process is repeated with the Tr-B as the training set for GAN and SVM, and Tr-A as the validation set. As shown in Figure 4, the whole processes are repeated for each acoustic scene class. The amount of feature pool was approximately 50 times that of the training DB (Tr-A or Tr-B), and amount of selected features was a similar to the training DB. In other words, the amount of the final augmented DB was about twice that of the original DB.

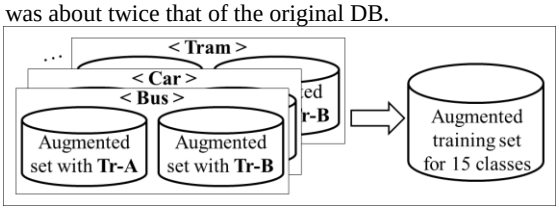

Figure 4: Final augmented training set for 15 classes

3. EXPERIMENTAL SETTINGS AND RESULTS

3.1. Input features and mid-level feature extractor configuration

For generating input features, audio signals sampled at 22.05 kHz sampling frequency were divided into 23.2ms frames with 512-size Discrete Fourier Transform (DFT). MFB and DFT spectrogram features were used as the input of the ASC feature extractor individually. Following [2], we used left, right, average and difference of both channel audio inputs. Total 4-types of sources were grouped into one DB set. We followed most of the details in neural network of [1]. The specific network architecture of the mid-level feature extractor is depicted in Table 1.

Table 1: The model specifications. (Batch size : 200 samples)

	MFB case	DFT case
Input	Input feature length : 1 [sec] / overlapping : 0.5 [sec]	
	[40 (feat.) x 42 (time-frame)]	[256 (feat.) x 42(time-frame)]]
LSTM #1 & 2	Hidden unit (300) / ReLU / Dropout (0.2)	Hidden unit (400) / ReLU / Dropout (0.2)
Conv. #1	4 x 4 (stride 1) / 10 filters / ReLU / Dropout (0.2)/ 2 x 2 max-pooling	16 x 8 (stride 1) / 10 filters /ReLU / Dropout (0.2) 2 x 2 max-pooling
Conv. #2	4 x 4 (stride 2) / 4 filters / ReLU / Dropout (0.2) 2 x 2 max-pooling	8 x 4 (stride 2) / 4 filters /ReLU / Dropout (0.2) 2 x 2 max-pooling
Mid-layer for **ASC feat.**	**Hidden unit (800)** / FCNN layer consists of two hidden layers with 400 ReLU units for each	
FCNN #1-3	Hidden unit (300) / Final Soft-Max layer (15) / Dropout (0.2)	

3.2. Generative adversarial network configuration

As mentioned in the introduction, various types of GANs have been widely researched. In this work, we do not focus on investigating more sophisticated sample generation methods. Instead, we use a basic GAN model [7, 9] to generate samples from the training data and show that these samples help to improve discriminative learning for the unseen validation data. In the GAN for 2-D images, the convolutional layer of DCGAN is generally used [4-6, 9] for 2-D matrix, but in this work, we used FCNN to process the feature vector (800 x 1 [dim.]) for simplicity. In order to help convergence of the discriminator, we added the normalized mean feature vector of the each class along with the random value as a GAN input. The specific network architecture of the GAN is depicted in Figure 5.

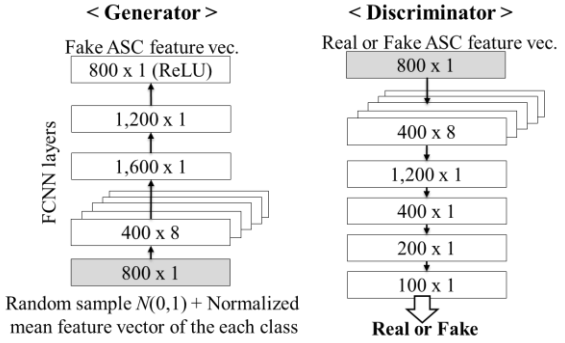

Figure 5: The neural network structure of GAN for each class

3.3. Classifier and late fusion

To assess the performance of the proposed DB augmentation, we conducted a number of experiments on the original DCASE 2017 development set and the augmented DB set, which consists of original samples and selected fake samples. As shown in the lower part of the Figure 1, SVM and FCNN were used as classifiers for the ASC feature inputs. The FCNN consists of 3-hidden layers with 300 hidden nodes and the SVM with a radial basis function kernel were used. For late fusion on the multiple classifier, we used linear logistic regression [2,17] on classification scores of DFT features with SVM and FCNN, and MFB features with SVM and FCNN in both cases of original and augmented DB. (8 =2 x 2 x 2=[feat. types] x [classifier types] x [DB types])

3.4. Results

We compared the average ASC accuracies over all scenes for the SVM and FCNN classifiers trained by the original DB and the augmented DB expended by the proposed framework. The file-based (10 [sec]) classification results are given in Table 2. For evaluation, the average waveforms on stereo audio input were used. Table 3 shows the class-wise classification accuracy on the fusion cases. As shown in Table 2, the proposed framework with the augmented DB set achieved higher accuracy than other cases. This can be interpreted that the proposed augmented DB set could infer properties of unseen DB and usage of the generated features could generalize or improve ASC performance. As given in Table 3, although the performances of the all classes were not improved by the proposed method, but the overall average accuracy outperformed the conventional approaches.

Table 2: Comparing the performance of the conventional and the proposed method (average accuracy on 4-fold validation)

Avg. acc. [%]	with original development set				with augmented set			
	DFT-FCNN	MFB-FCNN	DFT-SVM	MFB-SVM	DFT-FCNN	MFB-FCNN	DFT-SVM	MFB-SVM
	75.4	75.1	78.2	79.3	83.2	83.7	81.6	85.6

Table 3: The class-wise accuracy comparison on the dev. set

Acc. [%]	Baseline [14]	Fusion w/o augmented DB case	Fusion on all cases
Beach	75.3	70.9	71.8
Bus	71.8	82.1	87.2
Café	57.7	71.8	87.2
Car	97.1	89.0	88.5
City	90.7	85.6	98.7
Forest	79.5	97.3	94.9
Groce.	58.7	83.3	79.5
Home	68.6	76.0	89.7
Lib.	57.1	82.0	96.2
Metro	91.7	90.7	84.6
Office	99.7	95.1	96.2
Park	70.2	69.9	71.8
Resid.	64.1	71.8	87.2
Train	58.0	71.8	82.1
Tram	81.7	84.6	91.0
Avg.	74.8	81.5	**87.1**

3.5. Submissions

The experiments shown in the Table 2-3 were conducted with the default setting of the DCASE 2017 development (4-fold cross validation). However, in order to reflect more information of the development set for the challenge submission, we conducted the additional ASC feature generation based on the various DB configurations, such as 2-fold, 3-fold and 8-fold frameworks. In particular, additional DB augmentation processing was conducted on similar class pairs, such as train/tram, home/library and park/residential area. We will analyze a quantitative relationship between DB configuration and performance for the future research, after ground truth of the evaluation DB is published.

4. CONCLUSION AND FUTUREWORK

In order to improve ASC performance, this paper proposed a framework to generate feature samples using GANs. The novel method of using SVM hyper-plane to select features for performance improvement was proposed. Based on the experimental result of DCASE 2017 development set, we confirmed that the usage of the generated features could improve ASC performance. GAN and Variational Auto Encoders (VAEs) have shown impressive performance improvements in some studies, but it is still difficult to generate suitable training samples without bias. In order to alleviate the issue, we used an iterative method with added random values in the generated samples to mitigate the issue of sample bias and over fitting. Nevertheless, further statistical considerations and additional quantitative experiments are needed for generalization of training from GAN generated samples.

5. ACKNOWLEDGMENT

This subject is supported by Korea Ministry of Environment (MOE) as "Public Technology Program based on Environmental Policy".

6. REFERENCES

[1] S. Bae, I. Choi and N. Kim, "Acoustic scene classification using parallel combination of LSTM and CNN", Proceedings of the Detection and Classification of Acoustic Scenes and Events 2016 Workshop (DCASE2016), 2016.

[2] H. E. Zadeh, et. al., "CP-JKU Submissions for DCASE-2016: a hybrid approach using binaural i-vectors and deep convolutional neural networks", IEEE AASP Challenge on DCASE 2016 technical reports, 2016.

[3] http://www.cs.tut.fi/sgn/arg/dcase2016/task-results-acoustic-scene-classification

[4] A. Odena, "Semi-supervised learning with generative adversarial networks", arXiv:1606.01583, 2016.

[5] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training gans", In proc. NIPS 2016, pp. 2234-2242, 2016.

[6] Z. Zheng, L. Zheng, and Y. Yang, "Unlabeled samples generated by gan improve the person re-identification baseline in vitro", arXiv preprint arXiv:1701.07717, 2017.
[7] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets,. In proc. NIPS 2014, pp. 2672-2680, 2014.
[8] S. Mun, S. Shon, W. Kim, D.K. Han and H. Ko, "Deep Neural Network based learning and transferring mid-level audio features for acoustic scene classification", 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp.796-800, 2017.
[9] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks", arXiv preprint arXiv:1511.06434, 2015.
[10] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, "Context encoders: Feature learning by inpainting", In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2536-2544, 2016.
[11] C. C. Hsu, et. al., "Voice conversion from unaligned corpora using variational autoencoding wasserstein generative adversarial networks", arXiv preprint arXiv:1704.00849, 2017.
[12] T. Kaneko, et. al., "Generative adversarial network-based postfilter for statistical parametric speech synthesis", In Proc. ICASSP 2017, pp. 4910-4914, 2017.
[13] S. Pascual, S, A. Bonafonte and J. Serrà, "SEGAN: Speech Enhancement Generative Adversarial Network", arXiv preprint arXiv:1703.09452., 2017.
[14] http://www.cs.tut.fi/sgn/arg/dcase2017/.
[15] M. Oquab et. al., "Learning and transferring mid-level image representations using convolutional neural networks", In Proceedings of the IEEE Conference on computer Vision and Pattern Recognition (CVPR), pp. 1717-1724, 2014.
[16] C. Cortes and V. Vapnik, "Support-vector networks", *Machine learning*, vol. 20, no.3, pp. 273-297, 1995.
[17] N. Br¨ummer, "Focal multi-class: Toolkit for evaluation, fusion and calibration of multi-class recognition scorestutorial and user manual", Software available at http://sites.google.com/site/nikobrummer/focalmulticlass, 2007