

EXPERIMENTS ON THE DCASE CHALLENGE 2016: ACOUSTIC SCENE CLASSIFICATION AND SOUND EVENT DETECTION IN REAL LIFE RECORDING

Benjamin Elizalde¹, Anurag Kumar¹, Ankit Shah², Rohan Badlani³, Emmanuel Vincent⁴, Bhiksha Raj¹, Ian Lane¹

¹Carnegie Mellon University, Pittsburgh, PA, USA, ⁴Inria, F-54600 Villers-lès-Nancy, France

²NIT Surathkal, India, ³BITS, Pilani, India

bmartin1,alnu@andrew.cmu.edu, rohan.badlani,ankit.tronix@gmail.com,

emmanuel.vincent@inria.fr, bhiksha@cs.cmu.edu, lane@cmu.edu

ABSTRACT

In this paper we present our work on Task 1 Acoustic Scene Classification and Task 3 Sound Event Detection in Real Life Recordings. Among our experiments we have low-level and high-level features, classifier optimization and other heuristics specific to each task. Our performance for both tasks improved the baseline from DCASE: for Task 1 we achieved an overall accuracy of **78.9%** compared to the baseline of **72.6%** and for Task 3 we achieved a Segment-Based Error Rate of **0.76** compared to the baseline of **0.91**.

Index Terms— audio, scenes, events, features, segmentation, DCASE, bag of audio words, GMMs, sound event detection, acoustic scene classification

1. INTRODUCTION

Audio plays a critical role in understanding the environment around us. This makes audio content analysis research important for tasks related to multimedia [1, 2], and human computer interaction [3, 4] to mention a pair. However, unlike the field of computer vision which has a variety of standard publicly available datasets such as Imagenet, audio event/scene analysis lacks such large dataset. This makes it difficult to compare different approaches and establishing the state of art. The second iteration of DCASE [5], occurring in 2016, offers an opportunity to compare approaches on a standard public dataset. This edition it includes four different tasks: acoustic scene classification, sound event detection— real and synthetic audio, and audio tagging.

The state-of-the-art of the previous DCASE challenge, for both acoustic scenes [6–8] and sound event detection [6, 8, 9], attributed their success mainly to features and audio representations rather than classifiers. Hence, an important aspect in our work is to emphasize on classifier exploration along with features. In this paper we present our work performed on Task 1 and Task 3. We proposed a variety of methods for both tasks and we obtained significant improvement over the baseline methods.

2. TASKS AND DATA

The goal of Task 1, Acoustic Scene Classification, is to classify a test recording into one of predefined classes that characterizes the environment in which it was recorded for example *park*, *home*, *office*. TUT Acoustic Scenes 2016 dataset is used for this task. It consists of recordings from various acoustic scenes. For each recording location, a 3-5 minute long audio recording was captured. The original recordings were then split into 30-second segments for the challenge. There are 15 acoustic scenes for the task.

Task 3, Sound Event Detection in Real Life Recordings, evaluates performance of sound event detection in multi-source condi-

tions similar to our everyday life. There is no control over the number of overlapping sound events at each time, not in the training nor in the audio data. TUT Sound Events 2016 dataset is used for Task 3, which consists of recordings from two acoustic scenes: *Home* and *Residential Area*. There are 18 selected sound event classes, 11 for Home and 7 for Residential Area.

3. TASK 1: ACOUSTIC SCENE CLASSIFICATION

From machine learning perspective, we treated Task 1 as a multi-class classification problem. The first step is to use a suitable method for characterizing acoustic scenes in the audio segments. An effective approach for characterizing audio events is bag-of-audio-words based feature representation [10], which is usually built over low-level features such as MFCCs. Acoustic scenes, however, are more complex mixtures of different audio events and a more robust representation is required. To obtain a more robust representation we use Gaussian Mixture Models (GMMs) for feature representations of audio segments. Broadly, we employed two high-level feature representations to represent audio scenes. On the classification front we used Support Vector Machines (SVMs) as our primary classifier and in combination with other classifiers.

3.1. Feature Representations

Let D -dimensional MFCCs vectors for a recording be represented as \vec{x}_t , where $t = 1$ to T , T is the total number of MFCCs vectors for the recording. The major idea behind both high-level feature representation is to capture the distribution of MFCCs vectors of a recording. We will refer to these features as $\vec{\alpha}$ and $\vec{\beta}$ features and the sub-types will be represented using appropriate subscripts and superscripts.

The first step in obtaining high-level fixed dimensional feature representation for audio segments is to train a GMM on MFCC vectors of the training data. Let us represent this GMM by $\mathcal{G} = \{w_k, N(\vec{\mu}_k, \Sigma_k), k = 1 \text{ to } M\}$, where w_k , $\vec{\mu}_k$ and Σ_k are the mixture weight, mean and covariance parameters of the k^{th} Gaussian in \mathcal{G} . We will assume diagonal covariance matrices for all Gaussians and $\vec{\sigma}_k$ will represent the diagonal vector of Σ_k . Given the MFCCs vectors \vec{x}_t of a recording, we computed the probabilistic assignment of \vec{x}_t to the k^{th} Gaussian. These soft assignments are added over all t to obtain the total mass of MFCCs vectors belonging to the k^{th} Gaussian (Eq 1). Normalization by T is used to remove the effect of the duration of recordings.

$$Pr(k|\vec{x}_t) = \frac{w_k N(\vec{x}_t; \vec{\mu}_k, \Sigma_k)}{\sum_{j=1}^M w_j N(\vec{x}_t; \vec{\mu}_j, \Sigma_j)}, P(k) = \frac{1}{T} \sum_{t=1}^T Pr(k|\vec{x}_t) \quad (1)$$

The soft count histogram features referred to as $\vec{\alpha}$ is, $\vec{\alpha}^M = [P(1), \dots, P(k), \dots, P(M)]^T$. $\vec{\alpha}^M$ is an M -dimensional feature representation for a given recording. It captures how the MFCC vectors of a recording are distributed across the Gaussians in \mathcal{G} . $\vec{\alpha}^M$ is normalized to sum to 1 before using it for classifier training.

The next feature ($\vec{\beta}$), also based on the GMM \mathcal{G} , tries to capture the actual distribution of the MFCC vectors of a recording. This is done by adapting the parameters of \mathcal{G} to the MFCC vectors of the recording. We employ maximum *a posteriori* (MAP) estimation to for the adaptation [11] [12]. Parameter adaptation for k^{th} Gaussian follows the following steps. First we compute,

$$n_k = \sum_{t=1}^T Pr(k|\vec{x}_t), \quad E_k(\vec{x}) = \frac{1}{n_k} \sum_{t=1}^T Pr(k|\vec{x}_t)\vec{x}_t, \quad E_k(\vec{x}^2) = \frac{1}{n_k} \sum_{t=1}^T Pr(k|\vec{x}_t)\vec{x}_t^2 \quad (2)$$

Finally, the updated mean and variances are obtained as

$$\hat{\mu}_k = \frac{n_k}{n_k + r} E_k(\vec{x}) + \frac{r}{n_k + r} \vec{\mu}_k \quad (3)$$

$$\hat{\sigma}_k = \frac{n_k}{n_k + r} E_k(\vec{x}^2) + \frac{r}{n_k + r} (\vec{\sigma}_k^2 + \vec{\mu}_k^2) - \hat{\mu}_k^2 \quad (4)$$

The relevance factor r controls the effect of the original parameters on the new estimates. We obtain 2 different feature representation using the adapted means ($\hat{\mu}_k$) and variances ($\hat{\sigma}_k$). The first one denoted by $\vec{\beta}^M$ is an $M \times D$ dimensional feature obtained by concatenating the adapted means $\hat{\mu}_k$ for all k , that is $\vec{\beta}^M = [\hat{\mu}_1^T, \dots, \hat{\mu}_K^T]^T$. In the second $\vec{\beta}$ features adapted $\hat{\sigma}_k$ are concatenated along with $\hat{\mu}_k$ to obtain a $2 \times M \times D$ dimensional features. This form of $\vec{\beta}$ features are denoted by $\vec{\beta}_\sigma^M$.

3.2. Classification

Once the feature representation for audio segments have been obtained, Task 1 essentially becomes a multi-class classification problem. Our primary classifiers are SVMs where we explore a variety of kernels. For the $\vec{\beta}$ features, we use Linear Kernel (LK) and RBF Kernel (RK). For soft-count histogram $\vec{\alpha}$ features we explore a panoply of kernels. Along with LK and RK we explored the following kernels.

- Exponential χ^2 Distance (ECK): the kernel is computed as $K(\vec{x}, \vec{y}) = \exp^{-\gamma D(\vec{x}, \vec{y})}$, where $D(\vec{x}, \vec{y}) = \sum_i (x_i - y_i)^2 / (x_i + y_i)$ is χ^2 distance.
- χ^2 Kernel (CK): In this case $K(\vec{x}, \vec{y}) = \sum_i \frac{2x_i y_i}{x_i + y_i}$
- Intersection Kernel (IK): $K(\vec{x}, \vec{y}) = \sum_i \min(x_i, y_i)$
- Exponential Hellinger Distance Kernel (EHK): $K(\vec{x}, \vec{y}) = \exp^{-\gamma D(\vec{x}, \vec{y})}$ where $D(\vec{x}, \vec{y}) = \sum_i (\sqrt{x_i} - \sqrt{y_i})^2$
- Hellinger Kernel (HK): $(\vec{x}, \vec{y}) = \sum_i \sqrt{x_i y_i}$

The details of these kernels can be found in [13–15]. For kernels where γ term appears, the optimal value of γ value can be obtained by cross validation over training data. However, setting γ equal to the inverse of average distance $D(\vec{x}, \vec{y})$ between training data points works well in general as well. We use [16] [17] for SVM implementation.

Finally, we have a classifier fusion step where we combined the output of the different classifiers. We combined multiple classifiers by taking prediction vote from each classifier and the final predicted class is the one which gets the maximum vote. We call it the *Fused Classifier* and we observed that the fused classifier can give significant improvement for several acoustic scenes.

Table 1: Task 1 Accuracy for different cases (Single Classifier)

M	$\vec{\alpha}^M$							$\vec{\beta}^M$		$\vec{\beta}_\sigma^M$	
	LK	RK	ECK	CK	IK	EHK	HK	LK	RK	LK	RK
64	62.8	60.6	66.2	66.3	66.0	64.7	65.3	76.8	76.6	75.5	76.7
128	63.6	62.3	67.5	67.1	66.4	67.4	66.5	76.5	75.3	77.5	77.5
256	63.9	63.9	67.3	67.8	66.5	68.7	67.7	76.5	71.9	76.6	75.9
512	65.0	62.9	67.8	67.8	67.1	68.9	69.3	76.4	72.2	76.2	75.9

Table 2: Overall Task 1 Accuracy (Fused Classifier)

Scene	Baseline					Proposed				
	Fold 1	Fold 2	Fold 3	Fold 4	Avg.	Fold 1	Fold 2	Fold 3	Fold 4	Avg.
Beach	84.2	66.7	78.9	47.4	69.3	100	71.4	89.5	52.6	78.4
Bus	68.4	65.0	100	85.0	79.6	68.4	50.0	100	95.0	78.4
Cafe/Restaurant	66.7	94.7	71.4	100	83.2	88.9	63.2	76.2	95.0	80.8
Car	70.0	89.5	89.5	100	87.3	80.0	100	100	100	95.0
City Center	83.3	73.7	89.5	95.5	85.5	88.9	84.2	100	95.5	92.1
Forest Path	57.1	100	66.7	100	81.0	81.0	100	100	100	95.2
Grocery Store	52.6	81.0	89.5	36.8	65	89.5	81.0	94.7	84.2	87.3
Home	100	55.6	95.0	77.8	82.1	100	61.1	80.0	44.4	71.4
Library	47.6	38.9	15.0	100	50.4	47.6	33.3	85.0	100	66.5
Metro Station	84.2	94.4	100	100	94.7	94.7	94.4	100	100	97.3
Office	100	100	94.4	100	98.6	78.9	100	72.2	83.3	83.6
Park	10.0	5.6	0	40.0	13.9	65.0	33.3	50.0	30.0	44.6
Residential	78.9	47.6	100	84.2	77.7	84.2	42.9	94.7	57.9	69.9
Train	16.7	31.6	30.4	61.1	34.9	50.0	63.2	34.8	88.9	59.2
Tram	88.9	88.9	63.6	100	85.3	83.3	88.9	63.6	100	84.0
Overall	67.2	68.9	72.3	81.9	72.6	80.0	71.1	82.7	81.8	78.9

3.3. Results

Our experimental setup with the folds structure, is same as the one provided by DCASE. We extracted 20 dimensional MFCC features using 30ms window and 50% overlap. MFCCs are augmented with their delta and acceleration features. For our final feature representation we experimented with 4 different values of GMM component size M , 64, 128, 256 and 512. The relevance factor r for $\vec{\beta}$ is set to 20. Due to space constraints we cannot present fold-and-scene specific results for all cases and hence overall accuracy for all 4 folds is shown. Table 1 shows overall accuracy results for different cases. The accuracy for the MFCC-GMM *baseline* method provided in the challenge is 72.6%.

We can observe from Table 1 that $\vec{\alpha}$ features in general do not perform better than the baseline method for any SVM kernel. However, $\vec{\beta}$ features clearly outperformed baseline method. In the best case, with $M = 128$ and $\vec{\beta}_\sigma^M$ our method outperformed the baseline by an absolute 5%.

Table 2 shows results for the fused classifiers. For the fusion step we did not consider classifiers built over $\vec{\alpha}$ since these classifiers are inferior compared to those using $\vec{\beta}$ features. We can observe that our proposed method beats the baseline method by an absolute **6.3%**. Moreover, for scenes such as *Park*, *Train*, *Library* where the baseline method gives very poor results, we improved the accuracy by an absolute **16 – 30%**. We also obtained superior overall accuracy on all folds which suggests that our proposed method is fairly robust. This is further supported by the fact that on DCASE evaluation set, We achieved an overall accuracy of **85.9%**.

4. TASK 3: SOUND EVENT DETECTION IN REAL LIFE RECORDINGS

Detection of sound events in scenes and long recordings have been treated as a multi-class classification problem before in [18–20] where a classifier is trained with the sound segments. For testing, the classifier outputs segment/frame-level predictions for all the classes. In order to follow a similar approach, first we wanted to analyze features' performance for sound events regardless of the scene. This way, we could have an intuition of performance on the harder scenario of Task 3 where not every segment of the scene corresponds to a labeled sound event.

Feature Type	Accuracy%	Classifier
MFCCs	67.7	Logistic Regression
GBFB	52.4	Gradient Boosting
SGBFB	61.5	Gradient Boosting
Scatnet	62.1	Random Forest
Stacked	66.68	Random Forest
Stacked + PCA	66.06	Random Forest

Table 3: Sound-event classification accuracy for different feature types using the 18 sounds and the 75 training - 25 testing ratio. Stacked included normalized MFCCs +SGBFB +Scatnet.

4.1. Features and Classifiers Optimization

For the features we tried the conventional MFCCs with standard parameters such as 12 coefficients plus energy, delta and double delta for a total of 39 dimensions. Moreover, we explored three features addressing the time-frequency acoustic characteristics. The Gabor Filter Bank (GBFB) in [21] have 2D-filters arranged by spectral and temporal modulation frequencies in a filter bank. The Separable Gabor filter bank (SGBFB) features extract spectro-temporal patterns with two separate 1D GBFBs, a spectral one and a temporal one. This approach reduces the complexity of the spectro-temporal feature extraction and further improves robustness as demonstrated in [22]. Both features have the default parameters from the toolbox¹ for a total dimension of 1,020 each. The Scatnet [23] features are generated by a scattering architecture which computes invariants to translations, rotations, scaling and deformations, while keeping enough discriminative information. It can be interpreted as a deep convolution network, where convolutions are performed along spatial, rotation and scaling variables. As opposed to standard convolution networks, the filters are not learned but are scaled and rotated wavelets. The features were extracted with a toolbox² using 0.25 second segments. The dimensionality of the three Scatnet components are 2, 84, 435 for a total of 521. Additionally, we included the normalized (mean and variance) Stacked (MFCCs+ SGBFB+ Scatnet) with PCA and also the normalized (mean and variance) Stacked without PCA. For the PCA we used Scikit’s [24] and used the full dimensionality of 1,580 as the number of input components and the resultant automatic reduction was 909 dimensions. For all the feature types and for the sake of avoiding the length variability of the temporal dimension, we averaged the vectors across time to end up with one single vector per sound event file.

Then, for the classifiers we considered Tpot [25], built on top of Scikit [24], which is a Python tool that automatically creates and optimizes machine learning pipelines using genetic programming. This toolbox (version 4) considers 12 classifiers such as Decision Tree, Random Forest, Xtreme Gradient Boosting, SVMs, K-Neighbors and Logistic Regression. The main Tpot parameter is “number of generations”, which corresponds to the number of iterations carried to tune the classifier, we set it to 15. An example of the best classifier for each feature type can be seen in Table 3. Interestingly, decision tree-based algorithms and logistic regression outperformed others like SVMs.

For our experiments, we extracted the 18 sound events from the two scenes using the annotations, and then we extracted different feature types from these isolated sounds. For each feature type experiment, the sound events’ feature files were fed to Tpot in a

¹<http://www.uni-oldenburg.de/mediphysik-akustik/mediphysik/downloads/gabor-filter-bank-features/>

²<http://www.di.ens.fr/data/software/scatnet/>

Feature Type	Accuracy%	Classifier
Home	56.4	Random Forest
Home + G	55.2	Random Forest
Home + G + P	55.7	Random Forest
Residential	53.3	Gradient Boosting
Residential + G	57.8	Decision Tree
Residential + G + P	56.7	Random Forest

Table 4: Sound-event classification accuracy using the DCASE set up with four-folds partitions. The inclusion of the [G]eneric class improved performance for both scenes, whereas the inclusion of the [P]erturbed audio improved only the Home performance.

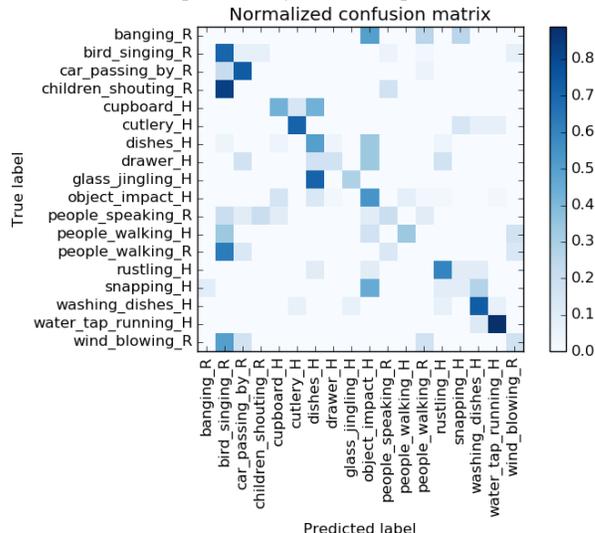


Figure 1: [H]ome and [R]esidential without the generic class. *Object impact* and *bird singing* capture most of ambiguities.

randomly selected ratio of 75% training and 25% testing, each set with different files. We kept the same partitions across our experiments for consistency. The performance was measured in terms of accuracy and is displayed in Table 3.

The features with the best performance were MFCCs with 67.7% and thus we keep them for our DCASE evaluation set up. The other features have shown better results than MFCCs on audio classification, but it wasn’t the case for this particular dataset. Results for Scatnet was 62.1%, for GBF was 52.4%, and for SGBF was 61.5%. Moreover, the two normalized stacked features performed almost as good as MFCCs with 66.68% for the stacked without PCA and 66.08% for the stacked with PCA. In principle the stacked version contains more information about the acoustics and thus they were expected to perform better. Nevertheless, they didn’t outperform MFCCs which is designed for speech and focus on lower frequencies rather than on a wider frequency range. We cannot draw a fundamental conclusion on the performance of these features for sound event classification since the amount of data and classes are determinant.

4.2. Inclusion of Generic Sound Event Class

In the annotated scenes, not every segment of audio corresponds to a labeled sound. Hence, it cannot be assured that any of our sound event classes have to be present on every test segment. To handle out-of-vocabulary segments, we proposed a generic sound event class.

For the first experiment we wanted to analyze the impact of the

generic class together with the 18 sounds in the multi-class classification set up described in Section 4.1 using MFCCs. To create such class, we used the sound events annotations and trimmed out the audio between the labeled segments, which are unlabeled. Then, we randomly selected from both scenes, 60 audio files which is about the average number of sound event samples per class. In order to visualize the performance, we included the normalized confusion matrices (CMs) in Figures 1 and 2. The accuracy performance without the generic class was 67.7% and with the generic class was 60.94%. The performance dropped with the inclusion of the new class, but we can also observe how although the generic class shared the background acoustics with the other sound classes, it didn't significantly ambiguous with them.

The second set of experiments used the DCASE setup of separate scenes and four folds, and utilized the sound events with and without the generic class, but this time the generic class will have files particular to the scene. The results can be seen in Table 4 showing benefit of including the generic class. Moreover, the CMs not included due to space limitations, had cleaner diagonals. The reasons for performance improvements on the DCASE set up are suggested by the utilization of less sound classes, which reduces class ambiguity. The utilization of the generic class built with same-scene files as opposed to a mix of both scenes. As well as the optimization per scene of the classifier using Tpot.

4.3. Generation of Data Through Perturbation

The scarcity of labeled data per event is a common issue as discussed in [26, 27]. Annotations are costly, sounds don't occur with the same frequency and in general it's hard to capture enough variations of the same sound to train robust models. To address this problem, multiple techniques have been explored in the literature such as perturbation of the audio signal as in [28, 29]. The authors presented multiple types of perturbations resulting in improvements of speech separation. For Task 3 we performed time-based perturbation by speeding up and slowing down the sound event samples. We empirically analyzed multiple combinations of speed up-down values for different events. We concluded that speeding up more than 30% the original signal resulted in unintelligible audio and speeding down the signal more than 100% would be unlikely to occur. The range included 13 different speed values and the original version.

The set of experiments used the DCASE setup of separate scenes and four folds, and utilized the time-based perturbed audio. For training, we added to the original files the 13 versions of the perturbed audio files, whereas for testing, the set remained intact. The results can be seen in Table 4, where the performance for *Home* improved, but not for *Residential*. Thus, we decided to use perturbation for the DCASE evaluation.

4.4. Sound Event Detection and Submission Systems

For Task 3, we used the DCASE setup of separate scenes and four folds in a similar setup as the experiments from Table 4. For each scene, we extracted the sound events from the recordings using the annotations from the train set, followed by the extraction of MFCCs features. After, we trained the Tpot optimized multi-class classifier with the event samples. For testing, instead of using sound event files only, we segmented the scene recordings from the test set into one-second consecutive segments. This number was selected due to the metric schema of the DCASE evaluation, which considers one-second segments. After, we extracted audio features from the test segments and evaluate them with the classifier to obtain scores for each trained sound event class. The label corresponding to the highest score was chosen for the segments and then were written down

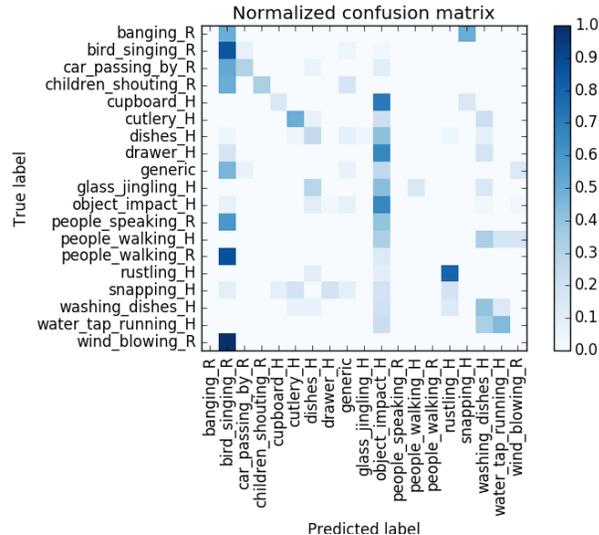


Figure 2: [H]ome and [R]esidential with the generic class. Although this class shared the background acoustics with the 18 sounds, it didn't cause major confusion.

Acoustic Scene	SBER	F-score
Home	1.05	25.1
Home + G	0.91	23.7
Home + G + P	0.9	24.7
Residential	0.64	54.6
Residential + G	0.72	45.9
Residential + G + P	0.63	52.2

Table 5: Our Segment-based Error Rate, using [G]eneric and [P]erturbation, outperformed the baseline.

into the DCASE format output file and fed to the official scoring scripts³ along with the ground truth to compute performance.

We utilized the pipeline for three experiments, without generic class & without perturbation, with generic class & without perturbation and with generic class & with perturbation. The results using the development-test set are shown in Table 5. The inclusion of the generic class and the perturbation outperformed the baseline method by a significant margin for both *Home* and *Residential* scenes. Our submission consisted on the runs using G and G+P but using the evaluation set. The eval results were SBER of 0.9613 and Fscore of 33.6% given by the G+P version.

5. CONCLUSION

In this paper we showed different approaches for both acoustic scene classification (Task 1) and sound event detection (Task 3) of the 2016 DCASE challenge. On both tasks we were able to obtain significant improvement over the baseline method. For Task 1 we observed that the β features performed much better than α features. Although, linear and RBF kernels with β features can outperform the baseline by considerable margin on its own, we make note of the fact that a multiple classifier system can give further improvements. For Task 3, we tested different features and classifiers and significantly improved the baseline. Moreover, we explored a way of handling out-of-vocabulary sound segments with the generic class and the inclusion of perturbed audio to add robustness.

³<http://www.cs.tut.fi/sgn/arg/dcase2016/sound-event-detection-metrics>

6. REFERENCES

- [1] H. Cheng, J. Liu, S. Ali, O. Javed, Q. Yu, A. Tamrakar, A. Divakaran, H. S. Sawhney, R. Manmatha, J. Allan *et al.*, “Sri-sarnoff aurora system at trecvid 2012: Multimedia event detection and recounting,” in *Proceedings of TRECVID*, 2012.
- [2] L. Jiang, S.-I. Yu, D. Meng, Y. Yang, T. Mitamura, and A. G. Hauptmann, “Fast and accurate content-based semantic search in 100m internet videos,” in *Proceedings of the 23rd ACM international conference on Multimedia*. ACM, 2015.
- [3] S. Chu, S. Narayanan, C.-C. J. Kuo, and M. Mataric, “Where am i? scene recognition for mobile robots using audio features,” in *2006 IEEE ICME*. IEEE, 2006, pp. 885–888.
- [4] M. Janvier, X. Alameda-Pineda, L. Girin, and R. Horaud, “Sound-Event Recognition with a Companion Humanoid,” in *Humanoids 2012 - IEEE International Conference on Humanoid Robotics*. Osaka, Japan: IEEE, 2012, pp. 104–111.
- [5] A. Mesaros, T. Heittola, and T. Virtanen, “TUT database for acoustic scene classification and sound event detection,” in *24th European Signal Processing Conference 2016 (EU-SIPCO 2016)*, Budapest, Hungary, 2016.
- [6] G. Roma, W. Nogueira, P. Herrera, and R. de Boronat, “Recurrence quantification analysis features for auditory scene classification,” *IEEE AASP Challenge*, Tech. Rep., 2013.
- [7] A. Rakotomamonjy and G. Gasso, “Histogram of gradients of time-frequency representations for audio scene classification,” *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 23, no. 1, pp. 142–153, 2015.
- [8] J. Schröder, N. Moritz, M. R. Schädler, B. Cauchi, K. Adiloglu *et al.*, “On the use of spectro-temporal features for the ieeeaasp challenge detection and classification of acoustic scenes and events,” in *2013 IEEE WASPAA*. IEEE, 2013.
- [9] J. F. Gemmeke, L. Vuegen, P. Karsmakers, B. Vanrumste *et al.*, “An exemplar-based nmf approach to audio event detection,” in *2013 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*. IEEE, 2013, pp. 1–4.
- [10] F.-F. Li and P. Perona, “The perceived position of moving objects: Transcranial magnetic stimulation of area MT+ reduces the flash-lag effect,” in *IEEE CVPR*, vol. 2, 2005.
- [11] J. Gauvain and C. Lee, “Maximum a posteriori estimation for multivariate gaussian mixture observations of markov chains,” *Speech and audio processing, IEEE Trans. on*, 1994.
- [12] F. Bimbot *et al.*, “A tutorial on text-independent speaker verification,” *EURASIP journal on applied signal processing*, vol. 2004, pp. 430–451, 2004.
- [13] J. Zhang, M. Marszałek, S. Lazebnik, and C. Schmid, “Local features and kernels for classification of texture and object categories: A comprehensive study,” *International journal of computer vision*, vol. 73, no. 2, pp. 213–238, 2007.
- [14] A. Vedaldi and A. Zisserman, “Efficient additive kernels via explicit feature maps,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 3, pp. 480–492, 2012.
- [15] P. Li, G. Samorodnitsk, and J. Hopcroft, “Sign cauchy projections and chi-square kernel,” in *Advances in Neural Information Processing Systems*, 2013, pp. 2571–2579.
- [16] C.-C. Chang and C.-J. Lin, “LIBSVM: A library for support vector machines,” *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011.
- [17] R.-E. Fan, K. Chang, C. Hsieh, X. Wang, and C. Lin, “Liblinear: A library for large linear classification,” *The Journal of Machine Learning Research*, 2008.
- [18] B. Elizalde, M. Ravanelli, K. Ni, D. Borth, and G. Friedland, “Audio-concept features and hidden markov models for multimedia event detection.”
- [19] B. Elizalde and G. Friedland, “Lost in segmentation: Three approaches for speech/non-speech detection in consumer-produced videos,” in *2013 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2013, pp. 1–6.
- [20] B. Elizalde, G. Friedland, H. Lei, and A. Divakaran, “There is No Data Like Less Data: Percepts for Video Concept Detection on Consumer-Produced Media,” in *ACM International Workshop on Audio and Multimedia Methods for Large-Scale Video Analysis at ACM Multimedia*, 2012.
- [21] M. R. Schädler, B. T. Meyer, and B. Kollmeier, “Spectro-temporal modulation subspace-spanning filter bank features for robust automatic speech recognition,” *The Journal of the Acoustical Society of America*, pp. 4134–4151, 2012.
- [22] M. R. Schädler and B. Kollmeier, “Separable spectro-temporal gabor filter bank features: Reducing the complexity of robust features for automatic speech recognition,” *The Journal of the Acoustical Society of America*, 2015.
- [23] L. Sifre and S. Mallat, “Rotation, scaling and deformation invariant scattering for texture discrimination,” in *Proceedings of the IEEE CVPR*, 2013, pp. 1233–1240.
- [24] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, and others., “Scikit-learn: Machine learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
- [25] R. S. Olson, R. J. Urbanowicz, P. C. Andrews *et al.*, “Automating biomedical data science through tree-based pipeline optimization,” in *Proceedings of the 18th European Conference on the Applications of Evolutionary and Bio-inspired Computation*, ser. Lecture Notes in Computer Science. Springer-Verlag, 2016.
- [26] A. Kumar and B. Raj, “Audio event detection using weakly labeled data,” in *24th ACM International Conference on Multimedia*. ACM Multimedia, 2016.
- [27] —, “Weakly supervised scalable audio content analysis,” in *2016 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2016.
- [28] J. Chen, Y. Wang, and D. Wang, “Noise perturbation improves supervised speech separation,” in *International Conference on Latent Variable Analysis and Signal Separation*. Springer, 2015, pp. 83–90.
- [29] N. Kanda, R. Takeda, and Y. Obuchi, “Elastic spectral distortion for low resource speech recognition with deep neural networks,” in *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on*. IEEE, 2013, pp. 309–314.