DNN BASED MULTI-LEVEL FEATURE ENSEMBLE FOR ACOUSTIC SCENE CLASSIFICATION

Jee-weon Jung, Hee-soo Heo*, Hye-jin Shim, and Ha-jin Yu*†

School of Computer Science, University of Seoul, South Korea

ABSTRACT

Various characteristics can be used to define an acoustic scene, such as long-term context information and short-term events. This makes it difficult to select input features and pre-processing methods suitable for acoustic scene classification. In this paper, we propose an ensemble model that exploits various input features in which the strength for classifying an acoustic scene varies: i-vectors are used for segment-level representations of long-term context, spectrograms are used for frame-level short-term events, and raw waveforms are used to extract features that could be missed by existing methods. For each feature, we used deep neural network based models to extract a representation from an input segment. A separated scoring phase was then exploited to extract class-wise scores on a scale of 0 to 1 that could be used as confidence measures. Scores were extracted using Gaussian models and support vector machines. We tested the validity of the proposed framework using task 1 of detection, and classification of acoustic scenes and events 2018 dataset. The proposed framework had an accuracy of 73.82% for the pre-defined fold-1 validation setup and 74.8% for the evaluation setup which is 7th in team ranking.

***Index Terms*—** Acoustic scene classification, DNN, raw waveform, i-vector

1. INTRODUCTION

There is an increasing demand for acoustic scene classification (ASC), a task that can be applied in various machines and intelligent systems. Three noticeable characteristics can be observed by analyzing the past editions of detection and classification of acoustic scenes and events (DCASE) competitions: (a) deep neural networks (DNNs) are mainly used with various architectures, (b) various features such as spectrograms, Mel frequency cepstral coefficients (MFCCs), and constant Q cepstral coefficients (CQCCs) [1] are used , and (c) ensemble of two or more classifiers are used with majority voting or score-sums.

Despite this active research, choosing appropriate features for ASC tasks remains difficult. One of the main factors complicating this problem may be the fact that different features are appropriate for representing each scene in an ASC task. For example, segment-level features such as i-vectors may be useful for classifying scenes where the characteristics appear over a long period of time. Frame-level features such as spectrograms can be used to classify scenes where events occur in a particular frequency band at short intervals. To consider the different characteristics that can define an acoustic scene, we trained DNNs that input each feature and agglomerate the results. Additionally, raw waveforms without feature extraction techniques can be input into the DNN to extract features internally with respect to ASC tasks during the training phase. By directly using raw waveforms, the DNN is expected to find most appropriate features for the target task.

Another problem is that the two methods most frequently used in ensembles of the DNNs (majority voting and score-sum) do not include confidence measures. In majority voting, the output of classifiers are voted, meaning that the precision ($\frac{true\ positive}{true\ positive + false\ positive}$) of the individual class of each system is not considered. Score-sum of DNN output layer uses a softmax activation as confidence score. This neglects the precision of classification on each class but also considered not ideal because in the case of softmax outputs, scores can be poorly calibrated [2]. Therefore, we added a separate scoring phase to calculate calibrated scores from trained DNNs [3].

In this paper, we make the following contributions:

1. Exploit various features including raw waveform that can be more useful for classifying acoustic scenes.
2. Train Gaussian models and support vector machines that inputs the output of DNN's code layer and extract scores with confidence.

Specifically, three features are individually studied for ASC task. The first feature is i-vector [4], a segment-level low dimensional representation, known to be suitable for ASC tasks. The second feature is a spectrogram, which is widely used for ASC task with convolutional neural networks (CNNs) [5], [6]. The last feature is a raw waveform, which is directly input into a DNN. We hypothesize that segment-level i-vectors can detect scenes using long-term context information, frame-level spectrogram can detect scenes involving short-term events, and raw waveforms can be used to find useful features for classifying acoustic scenes using DNN training. Single Gaussian models and support vector machines (SVMs) [7] use the outputs of DNN's code layers as input and are used as back-end classifiers to obtain confidence score for each class given an embedding. Final score is derived through score fusion using confidence scores. The overall proposed framework is depicted in Figure 1.

The remainder of this paper is organized as follows. Section 2 describes the three DNNs with different features, the back-end classifiers used in this study, and the ensemble methodology. The experimental settings and system specifications are presented in Section 3 with experimental results. The paper is concluded in Section 4.

2. SYSTEM DESCRIPTION

In this section, we describe the three systems in the ensemble according to their input features, the back-end classifiers used for scor-

*These authors have equal contribution.

† Corresponding author.

This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Science, ICT & Future Planning(2017R1A2B4011609)

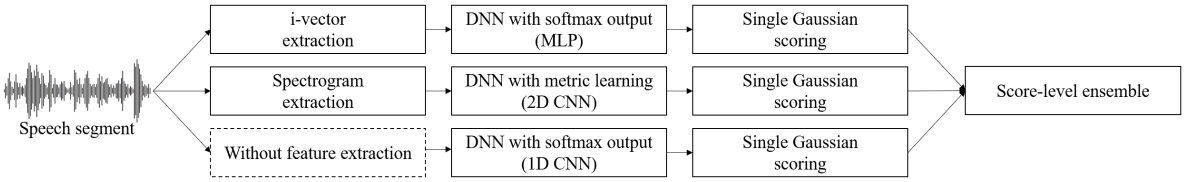

Figure 1: Illustration of the overall framework.

ing, and the ensemble methodologies.

2.1. i-vector based system

An i-vector (identity vector) is a low-dimensional representation of a given segment using factor analysis [4]. Regardless of the length of a given segment, one vector with fixed dimensionality is extracted. Originally, i-vectors were proposed for speaker verification, but in previous DCASE challenges, i-vectors have also performed well on ASC task [8, 9]. In this study, we used i-vectors as one of our input features, expecting that the segment-level representations would appropriately classify acoustic scenes defined by long-term contexts. The i-vector based DNNs is trained using a supervised training scheme with categorical cross-entropy objective functions and softmax activation.

2.2. Spectrogram based system

Spectrograms are widely used in audio signal processing systems, including speech recognition and speaker recognition. We hypothesized that this frame-level feature could be used to detect events occurring in a specific frequency band, and could therefore contribute to improving performance on ASC task. We used max feature map (MFM) based 2D CNN architecture to embed the spectrogram extracted from each segment [10]. In the MFM based architecture, instead of activation functions such as rectified linear units, a max operation is applied to multiple feature maps to calculate the output of each layer. In this study, we varied the filter sizes, expecting that the appropriate filter size would be found in the DNN training process through competition between filters.

The spectrogram-based system is trained using a metric learning scheme instead of conventional supervised training with softmax activation output layer. This learning scheme inputs two or more samples and trains the DNN to simultaneously decrease similarities between samples from different classes (= negative similarity) and increase similarities between samples from identical classes (= positive similarity). The cosine similarities are calculated between DNN embeddings at the code layer. Additionally, it has been shown that for performing the training of a DNN, it is more efficient for generalization to use similarities between an embedding and an average class embedding [11]. Therefore, the network is trained to minimize the loss defined by equation (1)

$$\mathcal{L} = \frac{1}{N_c}\sum_{i}^{N_c}\sum_{j\neq i}^{N_c}(CS(\boldsymbol{e}_i, \boldsymbol{m}_j) - CS(\boldsymbol{e}_i, \boldsymbol{m}_i)), \quad (1)$$

where, N_c is the number of classes, $\boldsymbol{e}_i$ is the embedding of a sample from the $i'th$ class, $\boldsymbol{m}_i$ is the average embedding of the $i'th$ class, and $CS(\cdot)$ is a cosine similarity operation between two embedding vectors. Figure 2 shows the process for calculating the positive and negative similarity samples defined in equation (1), based on ten classes. However, in datasets where the number of classes is small while the number of samples in each class is large, repeatedly calculating the average embeddings during the training process causes a large overhead.

Therefore, we calculated the average embeddings of each class at the beginning of the training as the centroid of each class and update it as the DNN training proceedes. The average embedding $\boldsymbol{m}_i^t$ of the $i'th$ class at time t is updated using equation (2)

$$\boldsymbol{m}_i^t = \alpha \boldsymbol{m}_i^{t-1} + (1-\alpha)\hat{\boldsymbol{m}}_i^t, \quad (2)$$

where,$\hat{\boldsymbol{m}}_i^t$ is the average embedding of class i calculated for each mini-batch, and α is momentum value which define a ratio between $\boldsymbol{m}_i$ and $\hat{\boldsymbol{m}}_i$.

Negative sampling is a technique that can effectively improve the performance of metric learning by searching hard negative cases [12]. In negative sampling, rather than using all samples, loss is calculated using samples that are relatively difficult to classify. However, negative sampling is time-consuming, and generally requires another classifier such as SVM solely for this operation. Instead of negative sampling, we used the modified loss shown in equation (3)

$$\mathcal{L}_{max} = \frac{1}{N_c}\sum_{i}^{N_c}\max_{\{0\leq j\leq N_c-1, j\neq i\}}(CS(\boldsymbol{e}_i, \boldsymbol{m}_j) - CS(\boldsymbol{e}_i.\boldsymbol{m}_i)), \quad (3)$$

In equation (3), positive similarities are used in the same way as the conventional loss defined in equation (1). On the other hand, only one of the $N_c - 1$ negative similarities is used to calculate loss, selected through max operations. Figure 3 shows an example of the operation of equation (3): the training process of $\mathbf{e}_1$ in which similarity with $\mathbf{m}_1$, the centroid of the same class, increases, and the similarity with $\mathbf{m}_2$, the centroid of the class that is most hard to classify, decreases. With such modifications, we expected that the DNN would be trained to better discriminate acoustic scenes with similar characteristics.

2.3. Raw waveform based system

Recently, promising results have been observed with DNNs that directly input raw waveforms. Such DNNs have been proposed for use with various tasks [13, 14, 15]. Through the visualization of raw-waveform-based DNN models, it has been shown that the kernels of 1D convolutional layers are trained to detect specific frequency bands [14]. Many raw waveform systems aim to extract

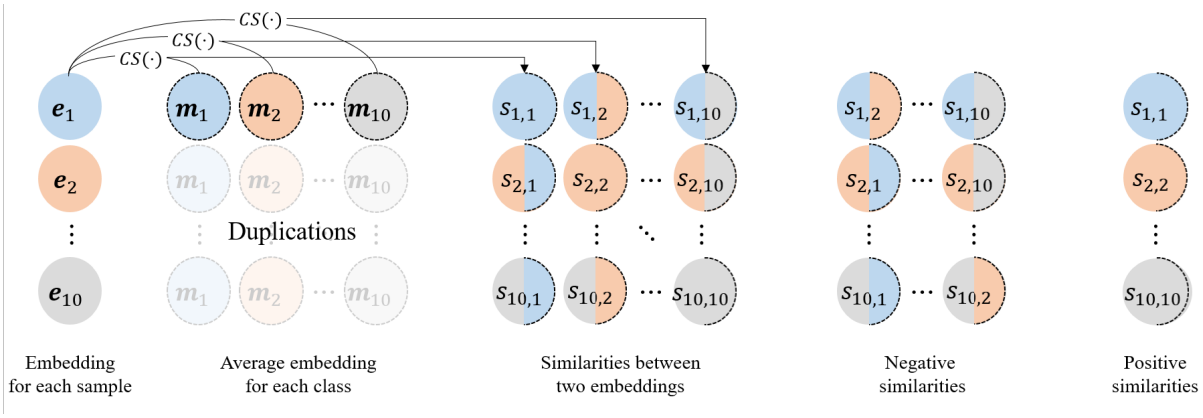

Figure 2: Concept illustration of the modified metric learning with learned mean embeddings of each class.

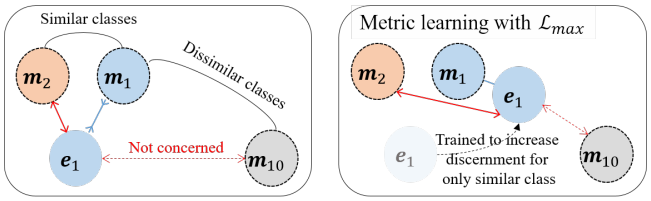

Figure 3: Concept illustration of equation (3) .

features that suit the objective defined by the loss function of DNN better than existing acoustic feature extracting techniques through extracting most useful frequency bands [14]. In this work, we use the RACNN-LSTM model proposed by Jung et al. [15] with a few modifications, considering the DCASE 2018 task 1 dataset. The raw waveform system that we used consists of convolutional blocks and fully connected layers: each convolutional block consists of a 1D convolutional layer, followed by batch normalization, rectified unit activation, and max pooling . The raw-waveform-based DNN is trained by supervised learning using a categorical cross-entropy loss function. Modifications and detailed descriptions of the raw-waveform-based system are present in Section 3.3.

2.4. Back-end scoring

Support vector machine (SVM) with RBF kernel and sigmoid kernel, single Gaussian model with diagonal and full covariance were used as back-end classifier. Classifiers were trained to discriminate acoustic scenes using DNN embeddings. In the spectrogram-based system, we used the code layer directly. The last hidden layer was used as the code layer for the i-vector and raw waveform systems. We expected that by using a back-end classifier for scoring instead of a softmax output, we could make the ensemble of multiple DNNs more efficient.

2.5. Ensemble method

Scores from each of the back-end scoring classifiers can be simply summed, because the scores already include the concept of confidence with a scale of zero to one. However, the different classifiers can have different discriminative powers for different acoustic scenes. To incorporate this concept, a precision vector is calculated based on the classification results for the validation dataset. The entries of a precision vector is the precision scores $\frac{true\ positive}{true\ positive + false\ positive}$ of each classifier for each class. Back-end classifier scores are multiplied by this precision vector before they are merged.

3. EXPERIMENTAL SETTINGS

Our experiments in this study used soundfile and scipy python modules for raw waveform and spectrogram extraction [16]. The Kaldi toolkit [17] was used for i-vector extraction. The Keras deep learning toolkit [18] with a tensorflow back-end [19, 20] was used for DNN training and decoding. The scikit-learn module was used for Gaussian model and SVM scoring [21].

3.1. Dataset

All experiments in this paper used task 1-a from the DCASE 2018 dataset [22]. Task 1-a in the DCASE 2018 dataset comprises 8,640 audio segments recorded in stereo at a 48 kHz sampling rate with 24 bit resolution and divided into 10 s lengths. Fourfold cross-validation was conducted using the provided meta data regarding recording locations. The development set and validation set do not use audio segments from identical locations. In this paper, we only report the accuracy of the first fold.

3.2. Feature configurations

We extracted i-vectors from a diagonal Gaussian mixture model (GMM) with 1024 components, trained with 60-dimensional MFCC features. A total variability matrix that can extract a 200-dimensional i-vector was trained for 10 iterations. Neither length normalization nor linear discriminant analysis were applied after i-vector extraction.

Spectrograms were extracted by shifting 30 ms windows by 10 ms. A spectrogram was represented by 721 coefficients for each window, and only 300 coefficients of low frequency bands were used; we empirically confirmed that low frequency bands are more

useful for ASC. Finally, a spectrogram of size 499×300 was extracted from each 10 s segment.

Stereo raw waveforms (with pre-emphasis) are used as input features to the DNN, resulting in feature shapes of $(48{,}000 \times 10, 2)$.

3.3. System configurations

The i-vector based DNN comprises 4 fully connected layers. In this system, the DNN acts only as a feature enhancer for the scoring step of the task, because the i-vector is already a sophisticated feature at the segment-level. The four fully connected layers each have 512 units, and L2 regularization is applied.

Spectrogram-based DNNs comprise two fully connected layers following three MFM layers. Fully connected layers contain 256 nodes activated by a leaky ReLU function [23]. L2 length normalization was applied to the output of the last fully connected layer following the work of Wan et al., who trained a DNN for speaker verification using metric learning [11]. The configuration of the MFM-based system is shown in Table 1. In each MFM layer, the output is calculated using the max operation between the feature maps generated by filters of different sizes. We simultaneously applied two types of pooling layers (max and average pooling) in the last CNN stage.

Table 1: Configuration of MFM based CNN system.

layer	output shape	kernel sizes
1^{st} MFM	499×300×32	5×5, 7×7, 9×9, 11×11
Max pooling	166×60×32	3×5
2^{nd} MFM	166×60×64	3×3, 5×5, 7×7, 9×9
Max pooling	55×12×64	3×5
3^{rd} MFM	55×12×64	3×3, 5×5, 7×7, 9×9
Max pooling	1×3×64	55×4
Average pooling	1×3×64	55×4
Concatenation	1×3×128	
Flatten	384	

Raw-waveform-based DNNs use the RACNN-LSTM model from Jung et al.'s work, with a few modifications [15]. Modifications include the following: the stride size of the strided convolutional layer was changed to 12 for a 48 kHz sampling rate, 256 kernels were used for the last convolutional layer, and stereo audio inputs were used instead of mono audio inputs.

3.4. Results

Experimental results for the provided fold 1 setup of the DCASE 2018 competition are presented in Table 2 in terms of classification accuracy. Each input feature is examined by four different classifiers, and the best results submitted to the DCASE 2018 competition are shown. The four columns of Table 2 each represent our submitted system for the DCASE 2018 competition, in which 'All' refers to the ensemble of single Gaussian and SVM classifiers.

Surprisingly, for the input features, raw waveforms performed the best, with an accuracy of 67.15 %. The three-feature ensemble increased accuracy more than 6 %. Although we do not show this result because of paper length limitations, the ensemble results for any two features improved performance in terms of classification accuracy. Therefore, we conclude that different features actually contribute to the ASC task based on the characteristics of each feature.

For back-end classifiers, we first compared the results of directly using softmax activation based classification to the results for separate scoring schemes using single Gaussian and SVM models. With the raw waveform as an input, conventional classification showed an accuracy of 64.71 %, while single Gaussian scoring and SVM scoring using the last hidden layer as code showed accuracies of 67.91 % and 66.56 %, respectively. Among the back-end classifiers, the accuracies of single Gaussian models were approximately 1 % higher, but noticeable differences were not measured. By applying a precision vector representing the accuracy of each acoustic scene system, we were able to improve performance when the precision vector was used with classifiers of the same type (e.g., diagonal Gaussian models and full Gaussian models). However, accuracy did not increase when the precision vector was used with different types of classifiers.

Table 2: Classification accuracy (%) for the individual systems and four-classifier ensemble system. The four columns indicate the four systems submitted to the DCASE task 1-a competition ('All w/o weight' is submission 1). 'w weight' refers to the case where classifier outputs were ensembled with the use of a precision vector. All refers to cases using two Gaussian and two SVM classifiers, Gaussian refers to cases using full and diagonal covariance classifiers, and SVM refers to cases using SVMs with RBF and sigmoid kernels.

system \ classifier	All w/o weight	All w weight	Gaussian w weight	SVM w weight
raw-waveform (val)	67.15	68.10	67.91	66.56
spectrogram (val)	66.24	66.20	66.44	66.44
i-vector (val)	63.74	63.93	65.17	63.66
Ensemble (val)	**73.82**	73.23	73.15	72.71
Ensemble (eval)	**74.8**	74.2	73.8	73.8

4. CONCLUSION AND FUTURE WORKS

Selecting appropriate features for each task is critically important for machine learning research. However, this is difficult because of the required domain expertise, such as knowledge regarding the characteristics of the input data and the understanding of the task to be performed. Segment-level i-vectors and frame-level spectrograms were used to detect both long-term contexts and short-term events, by training DNNs for each feature. Additionally, raw waveforms were used, with expectation that the kernel weights for 1D convolutional layers would be trained to extract the most discriminative features for each ASC task. We built an ensemble of DNNs with different input features, using score fusion on single Gaussian models and SVMs. An accuracy of 73.82 % and 74.8 % was shown for the DCASE 2018 task 1-a validation set and evaluation set, respectively.

In this study, we exploited multiple DNNs with different architectures, which respectively received different features. We then combined the results for each DNN. Training different types of features with a single DNN, however, may lead to the synergy of different features when training an integrated DNN. In the future, we plan to build a single integrated system that simultaneously receives multiple features. To achieve this type of system, we would need to simultaneously consider the various characteristics of different types of features.

5. REFERENCES

[1] M. Todisco, H. Delgado, and N. Evans, "Constant q cepstral coefficients: A spoofing countermeasure for automatic speaker verification," *Computer Speech & Language*, vol. 45, pp. 516–535, 2017.

[2] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," *arXiv preprint arXiv:1706.04599*, 2017.

[3] S. Park, S. Mun, Y. Lee, and H. Ko, "Score fusion of classification systems for acoustic scene classification," *IEEE AASP Challenge on Detection and Classification of Acoustic Scenes and Events (DCASE)*, 2016.

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

[5] Z. Weiping, Y. Jiantao, X. Xiaotao, L. Xiangtao, and P. Shaohu, "Acoustic scene classification using deep convolutional neural network and multiple spectrograms fusion," in *Detection and Classification of Acoustic Scenes and Events 2017 Workshop (DCASE2017)*, 2017.

[6] E. Fonseca, R. Gong, D. Bogdanov, O. Slizovskaia, E. Gómez Gutiérrez, and X. Serra, "Acoustic scene classification by ensembling gradient boosting machine and convolutional neural networks," in *Virtanen T, Mesaros A, Heittola T, Diment A, Vincent E, Benetos E, Martinez B, editors. Detection and Classification of Acoustic Scenes and Events 2017 Workshop (DCASE2017); 2017 Nov 16; Munich, Germany. Tampere (Finland): Tampere University of Technology; 2017. p. 37-41.* Tampere University of Technology, 2017.

[7] B. Schölkopf, A. J. Smola, *et al.*, *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2002.

[8] B. Elizalde, H. Lei, G. Friedland, and N. Peters, "An i-vector based approach for audio scene detection," *IEEE AASP Challenge on Detection and Classification of Acoustic Scenes and Events*, 2013.

[9] J. Jung, H. Heo, I. Yang, S. Yoon, H. Shim, and H. Yu, "Dnn-based audio scene classification for dcase 2017: Dual input features, balancing cost, and stochastic data duplication," in *Detection and Classification of Acoustic Scenes and Events*, 2017.

[10] X. Wu, R. He, Z. Sun, and T. Tan, "A light cnn for deep face representation with noisy labels," *arXiv preprint arXiv:1511.02683*, 2015.

[11] L. Wan, Q. Wang, A. Papir, and I. L. Moreno, "Generalized end-to-end loss for speaker verification," *arXiv preprint arXiv:1710.10467*, 2017.

[12] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 815–823.

[13] D. Palaz, M. Doss, and R. Collobert, "Convolutional neural networks-based continuous speech recognition using raw speech signal," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 4295–4299.

[14] J. Lee, J. Park, K. Kim, Luke, and J. Nam, "Sample-level deep convolutional neural networks for music auto-tagging using raw waveforms," *arXiv preprint arXiv:1703.01789*, 2017.

[15] J. Jung, H. Heo, I. Yang, H. Shim, and H. Yu, "A complete end-to-end speaker verification system using deep neural networks: From raw signals to verification result," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018.

[16] E. Jones, T. Oliphant, P. Peterson, *et al.*, "SciPy: Open source scientific tools for Python," 2001–, [Online; accessed today]. [Online]. Available: http://www.scipy.org/

[17] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, *et al.*, "The kaldi speech recognition toolkit," in *IEEE 2011 workshop on automatic speech recognition and understanding*, no. EPFL-CONF-192584. IEEE Signal Processing Society, 2011.

[18] F. Chollet *et al.*, "Keras," https://github.com/keras-team/keras, 2015.

[19] A. Martn, A. Ashish, B. Paul, B. Eugene, *et al.*, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," 2015. [Online]. Available: http://download.tensorflow.org/paper/whitepaper2015.pdf

[20] A. Martin, B. Paul, C. Jianmin, C. Zhifeng, D. Andy, D. Jeffrey, D. Matthieu, G. Sanjay, I. Geoffrey, I. Michael, K. Manjunath, L. Josh, M. Rajat, M. Sherry, M. G. Derek, S. Benoit, T. Paul, V. Vijay, W. Pete, W. Martin, Y. Yuan, and Z. Xiaoqiang, "Tensorflow: A system for large-scale machine learning," in *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, 2016, pp. 265–283. [Online]. Available: https://www.usenix.org/system/files/conference/osdi16/osdi16-abadi.pdf

[21] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[22] A. Mesaros, T. Heittola, and T. Virtanen, "A multi-device dataset for urban acoustic scene classification," 2018, submitted to DCASE2018 Workshop. [Online]. Available: https://arxiv.org/abs/1807.09840

[23] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. icml*, vol. 30, no. 1, 2013, p. 3.